

# **SwyftWare™**

*Using SwyftWare on the Apple //e and  
//c: a Guide, Glossary, and Reference  
Manual*

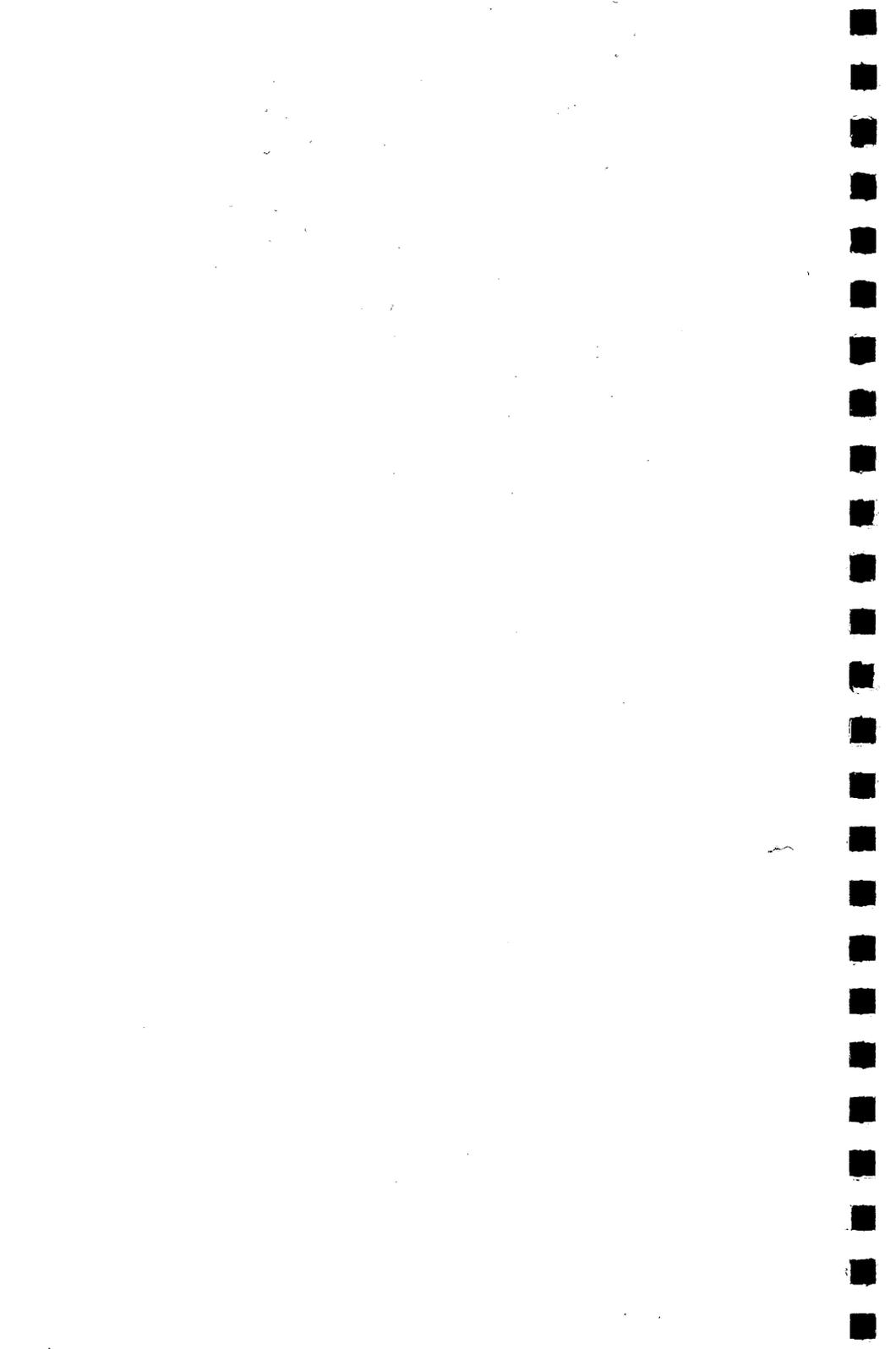
**Second Edition**

*SwyftWare by  
Jef Raskin, James Winter,  
Terry Holmes, Minoru Taoyama,  
Jonathan Sand, John Bumgarner,  
David Lavond, Paul Baker, Jim Straus,  
and Renwick Curry*

*Manual by  
David Alzofon and Jef Raskin*

**Information Appliance Inc.**

*1014 Hamilton Court, Menlo Park, CA 94025  
Telephone: (415) 328-5160  
Fax: (415) 321-0780 Telex: 821992*



## ***Dedication***

This manual is dedicated to all SwyftWare users. Unfortunately, it was written by all-too-human beings. We have tried to make it complete so that everything you need is included. This made it larger than we wanted. To save you time and to make the manual small and unimimidating, we tried to make it concise. This made it less explicit than we wanted. We have tried to make it enjoyable, but our sense of humor may not match yours. We have tried very hard to make it accurate, but are sure that an error or two has slipped in.

We know SwyftWare works. This manual was written and edited on the SwyftCard system running on Apple //e's. We printed the drafts with the PRINT command, and calculated various values with the CALC command. We used the DISK command and never lost any text. The SEND command was used to transmit the text to a typesetting system, which then made up the version you hold in your hand.

If you encounter a problem, or you have a suggestion that would improve the manual or the product, drop us a line. If you include a self-addressed stamped envelope we'll respond directly back to you.

— *Jef, David and Suzanne*

## Credits

*Movies have the fine tradition of giving credit (or blame, in the case of some) to nearly everybody who worked on the project. We can do no less:*

*David Alzofon wrote, tested, and edited the manuals and the on-line tutorial. Paul Baker honcho'd the parts and circuit boards through production.*

*Terry Holmes did the initial internal software design and the actual programming of the early versions of SwyftCard, and insisted on making everything run as fast as possible. Jonathan Sand transformed the prototype software into the final products, adding features and fine-tuning. Dr. David Lavond programmed during the final phase and completed the documentation.*

*Richard Krause designed the labels and did manufacturing coordination. Patti Okoniewski is the friendly voice you hear when you call us.*

*Jim Straus (who banished the company picnic and substituted an annual gourmet lobster fest — not at company expense) and Mino Taoyama\* contributed much to design and debugging; Mino's main contributions were the hardware design of the SwyftCard itself, and the design and implementation of the internal disk routines and ProDOS converter.*

*Ralph Voorhees\*, while mainly working on other projects, kept reminding us that our customers will not put up with the usual computer nonsense; Sharmon Scipio\* kept the paychecks coming and the books in order; John Bumgarner\* did some coding of the early versions and was a constant source of inspiration, as was the ever-effervescent Scott Kim.*

*Collaborators who are not currently employees include Dr. Ren Curry, Brian Howard\*, and Michael Raskin, who helped early on with the design and human factors testing. Ren's contribution to the cursor design was especially noteworthy. Deb Clifford worked on the early versions of the on-line tutorial; Charlie Springer contributed his technical know-how, and Cerafin Castillo was our electronics tech; Ron Genise\* did testing and teacher training. For a time Mike Scott, Sonya Hoag, Claire Llewelyn, Bana Witt, Camillo Santomero, and Tina Bracewell ably assisted us. Steve Bowling gave us his management and financial expertise when it was most needed. Bruce Cooper extensively tested an early prototype in a publication environment and provided helpful suggestions about formatting. Dr. Donna L. Cooper contributed the name "SwyftWare."*

*Mike Ivanitsky and Kevin Thurber tested printers, and Suzanne West of West and Moravec did the graphics.*

*A company is fueled by the money of its investors: we thank them all. The company would not have been possible without the legal advice of James E. Lewis\* and the work of corporate secretary Linda Blum\*. We also thank Gayla Newsome of Orange Micro in Santa Clara for allowing us to test the printer set-ups.*

*A special note of gratitude is extended to the many people who tested our systems, and to the San Jose and Fremont School Districts for letting us test SwyftCard in an educational environment.*

*The company grew from discussions between Dr. James Winter\* who invented the DISK command and contributed greatly to the design and implementation of the software, and Jef Raskin\*, who created the concepts that underlie SwyftCard and its implementation, founded Information Appliance, and who is now Chairman of the company.*

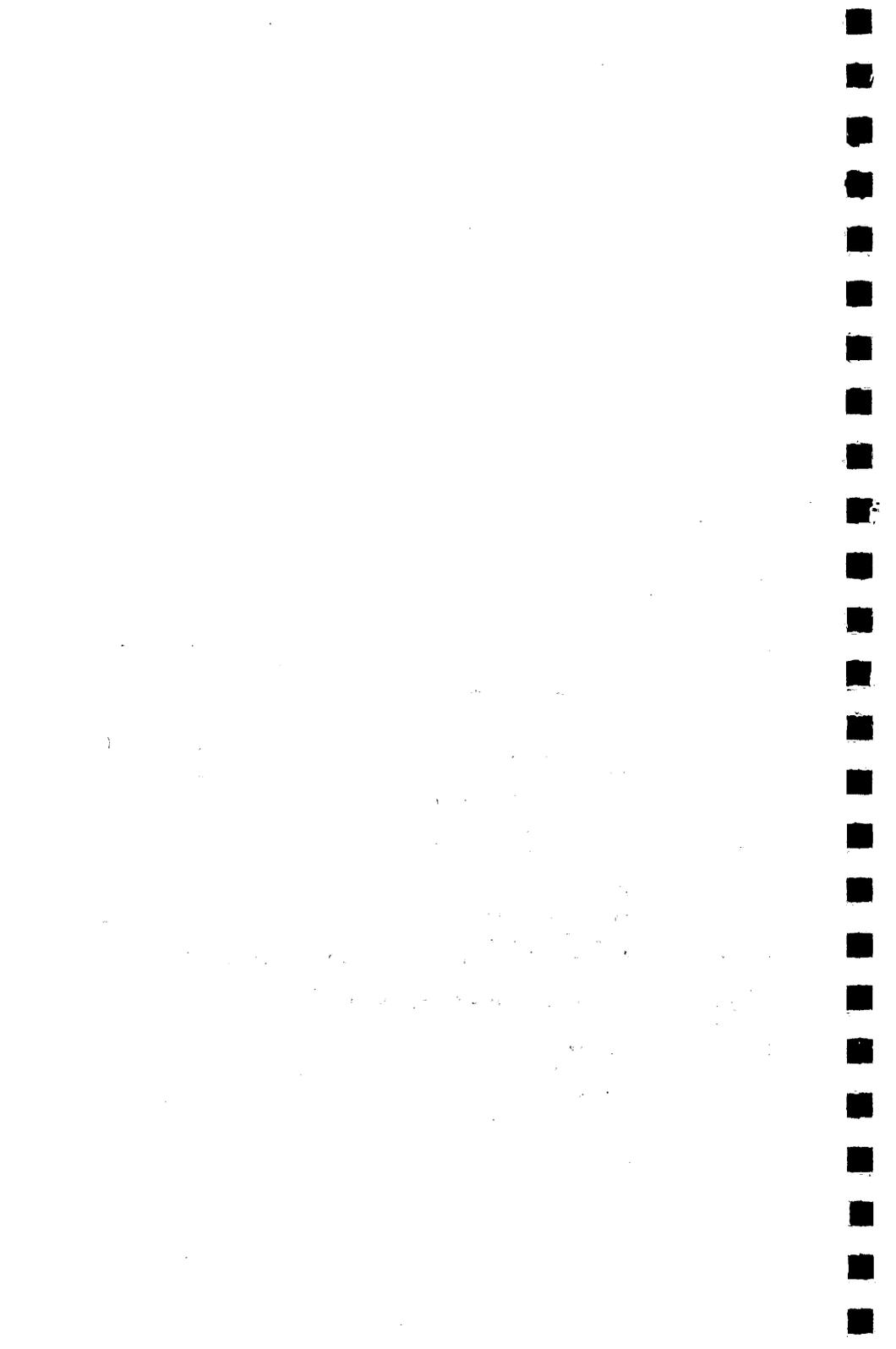
\*An asterisk marks a founder of Information Appliance Inc.

# The SwyftWare Manual

## Table of Contents

---

<b>Chapter I</b>		<b>Getting Started</b>	
		Introduction	1
		How to Learn SwyftWare	2
Part 1		Equipment Checklist for Apple //e Users	3
Part 2		Equipment Checklist for Apple //c Users	5
Part 3		Install the SwyftCard (//e Users Only)	7
Part 4		Label the Keys	9
Part 5		Start Up	10
Part 6		Starting Something of Your Own	13
<b>Chapter II</b>		<b>Editing</b>	
Part 7		Typing	15
Part 8		Pagination	21
Part 9		Cursor Motion	24
Part 10		Extending the Highlight	32
Part 11		DELETE	35
Part 12		The INSERT Command	37
<b>Chapter III</b>		<b>Saving and Calculating</b>	
Part 13		The DISK Command	39
Part 14		The CALC Command	51
Part 15		CALC Command Miscellany	53
Part 16		BASIC in the Swyft Environment	55
<b>Chapter IV</b>		<b>Printing and Communicating</b>	
Part 17		Printing	57
Part 18		Formatting the Printout	60
Part 19		Detailed Control of Your Printer	66
Part 20		The SEND Command (Electronic Communications)	79
<b>Chapter V</b>		<b>Problem Solving &amp; Information Retrieval</b>	
Part 21		Stopping	89
Part 22		Emergency Restart	91
Part 23		Bleeps	93
Part 24		Information Retrieval and Other Applications — a Few Ideas	98



---

**Appendices**

Appendix A	ProDOS Converter	102
Appendix B	Theory of Operation/Schematic	111
Appendix C	List of Features	118
Appendix D	Reassigning the Key Commands	121

---

<b><i>Glossary</i></b>	125
------------------------	-----

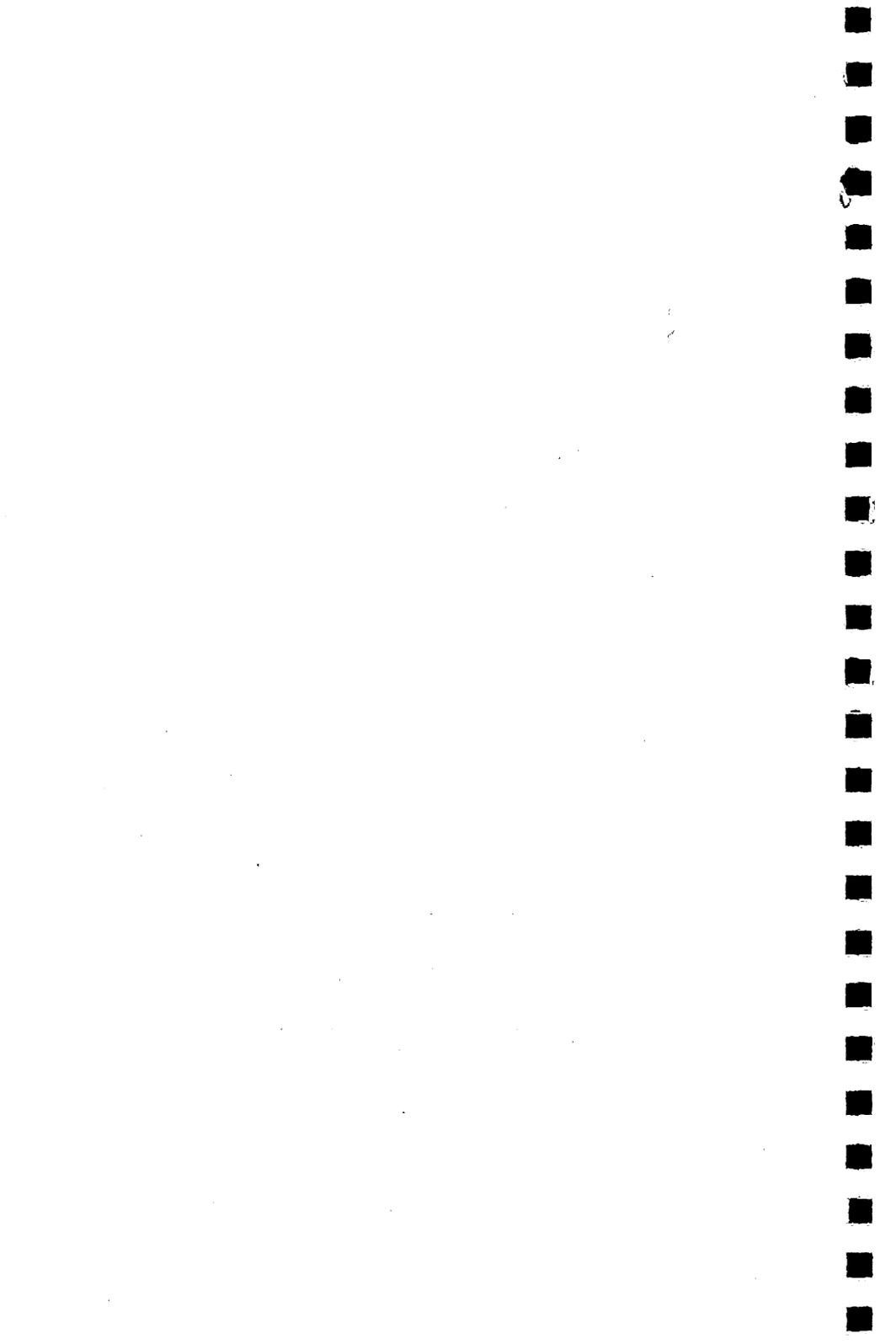
***Index******Reference Guide***



# **Chapter 1**

## **Getting Started**

	<i>Introduction</i>	1
	<i>How to Learn SwyftWare</i>	2
<i>Part 1</i>	<i>Equipment Checklist for Apple //e Users</i>	3
<i>Part 2</i>	<i>Equipment Checklist for Apple //c Users</i>	5
<i>Part 3</i>	<i>Install the SwyftCard (//e Users Only)</i>	7
<i>Part 4</i>	<i>Label the Keys</i>	9
<i>Part 5</i>	<i>Start Up</i>	10
<i>Part 6</i>	<i>Starting Something of Your Own</i>	13



## ***Introduction***

### ***SwyftWare™ Meets Apple™ \****

When you turn on the power to your computer, SwyftWare will be ready to help you

- produce written material
- store and find information (information retrieval\*)
- do arithmetic
- write short programs and, if you have a modem,
- send and receive information over phone lines

***Apple //e users:*** *You will not have to remove your SwyftCard in order to run most Apple software.*

### ***SwyftWare Is Not Just a Word Processor***

If it sometimes seems that it is, that's because nearly everything you do with a computer consists of typing to it and having it type back at you. With the exception of graphic or audible output, everything done with a computer is some form of word processing.

Much of this manual explains methods of typing and modifying text. This is a useful tool in its own right. But, as you can see from the list above, you can — without leaving SwyftWare — do calculations and (if you are so inclined) write programs; you can communicate with other SwyftWare-equipped Apples, any personal computer with telecommunications software, computer bulletin boards and commercial data services; and you can move information to and from and among disks.

SwyftWare is perfect for instantly retrieving correspondence, names, addresses, and telephone numbers. You can, for example, find a name anywhere in your text in the blink of an eye. The same goes for a phrase, a word, or a part of a word. And it doesn't matter to SwyftWare whether your text is neatly organized or a disorganized mess.

*\*If these or any other terms appearing in the manual are new to you, please consult the Glossary.*

## ***How to Learn SwyftWare***

You don't need to read this manual from cover to cover before you start. Just follow the instructions for setting up your equipment and then study the tutorial disk. If you are using SwyftDisk rather than SwyftCard, simply put the tutorial disk in the drive and turn on your computer. You'll learn the system faster if you just plunge in and try it.

The next few pages of the manual will get you set up and ready to do the tutorial. After you've done the tutorial (most people take about 90 minutes) you will know how to type and edit text, how to use SwyftWare's unique cursor, how to do information retrieval, and how to use each of SwyftWare's five commands.

The rest of the manual is for your reference. It covers the material presented in the tutorial and provides additional details, such as how to initialize your printer and alter the format of your printout. Keep the *Reference Guide* (back of the manual) by the computer for quick consultation.

## ***Part 1: Equipment Checklist for Apple //e Users***

### ***What You Need to Get Started***

- SwyftCard (or SwyftDisk)
- Apple //e (regular or enhanced)
- 80-column card (either standard or extended for SwyftCard, extended 80-column card required only for SwyftDisk)
- video monitor suitable for 80-column display
- disk drive with controller (you really need only one disk drive when using SwyftWare!)
- blank disks on which to save your work
- adhesive-backed labels for your Apple //e keyboard (these came in the SwyftWare package)

### ***What You Need for Printing***

To put your words on paper you will need a printer, printer interface, printer cable and, of course, some paper.

### ***What You Need for Telecommunications***

To send and receive information over telephone lines you will need an Apple Super Serial Card, a modem and modem cables.

### ***For ProDOS File Conversion***

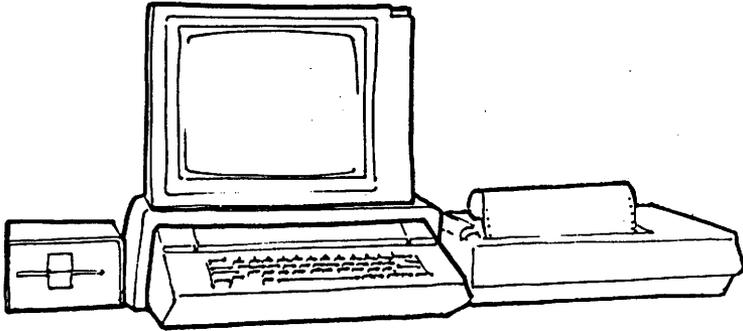
To convert SwyftWare text to a ProDOS text file and vice versa you will need the ProDOS Converter program on the flip side of the tutorial disk, a second disk drive or an extended 80-column card.

## ***Slot Assignments for Apple Cards***

If you're going to use a printer, a printer card must be in Slot 1; if you're going to use a modem, an Apple Super Serial Card must be in Slot 2; SwyftCard goes in Slot 3; the disk controller goes in Slot 6. Additional cards in other slots will usually not affect the operation of SwyftCard.

## ***Using SwyftWare With Other Cards***

If you have an accelerator card, SwyftCard may not work in your system. SwyftDisk will work, but you'll need an extended 80-column card. Other brands of extended 80-column cards will work with SwyftWare as long as they are Apple-compatible (that is, they don't require preboot disks to get started).

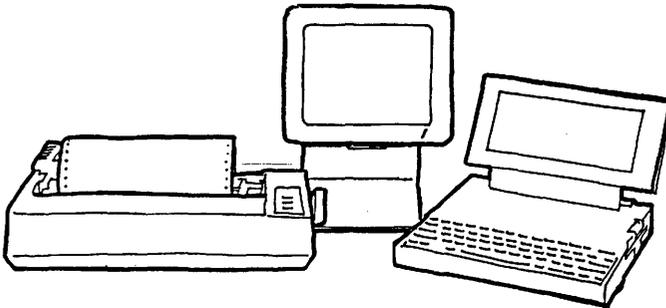


*What you need. A printer, modem and their interfaces are optional.*

## ***Part 2: Equipment Checklist for Apple //c Users***

### ***What You Need to Get Started***

- SwyftDisk
- Apple //c with power supply
- //c monitor and monitor cable or liquid crystal flat panel and cable
- blank disks on which to save your work
- adhesive-backed labels for your Apple //c keyboard (these came in the SwyftDisk package)



*Left to right: Printer, standard Apple //c monitor, and an Apple //c computer equipped with a C-VUE \* liquid crystal flat panel. This type of display is, of course, optional, as is the printer.*

*\*C-VUE Display Panel is a registered trademark of SWI International Systems.*

### ***What You Need for Printing***

To put your words on paper you will need a printer, printer cable, and, of course, some paper.

### ***What You Need for Telecommunications***

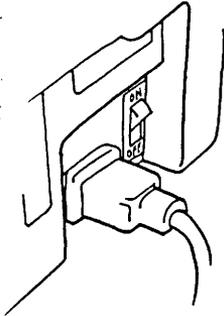
To send and receive information over telephone lines you will need a modem and modem cable.

### ***For ProDOS File Conversion***

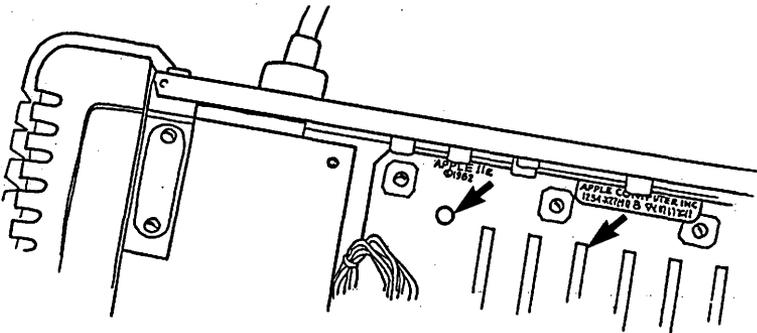
To convert SwyftWare text to a ProDOS text file and vice versa you will need the ProDOS Converter program on the flip side of the tutorial disk.

### **Part 3: Install the SwyftCard (//e Users Only)**

Switch off the power switch to your //e and remove the cover. Check the little red light in the back left corner — this light should be off if the power is off. Do not unplug the computer: the ground connection helps drain static electricity which could damage the computer, and there are no dangerous voltages on the inside of the computer when it is off.



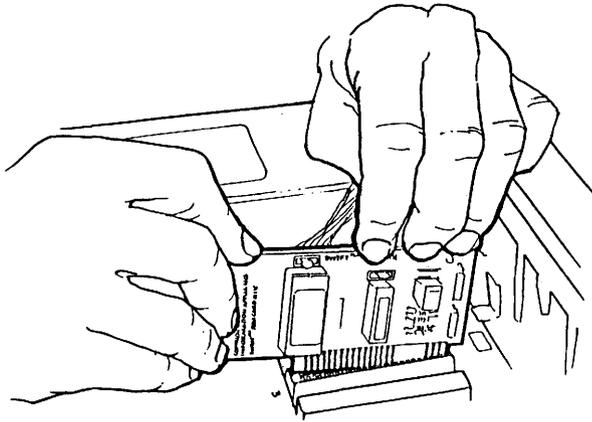
*Turn the power switch OFF, leave the system plugged in.*



*The power light inside the computer should be off.  
Note the location of Slot 3.*

Seat the fingers of the SwyftCard in Slot 3 (the fingers are the metal strips that look like the teeth of a gold comb). SwyftCard can only fit one way, with the long portion toward the keyboard.

Gently but firmly push the SwyftCard into the slot until it touches bottom. With the SwyftCard snugly installed, replace the cover and push down in back to latch it.



*Installing SwyftCard in Slot 3*

Install other cards according to the instructions in their manuals.



## **Part 5: Start Up**

### **How to Start the Tutorial**

Place the tutorial disk in the disk drive (the disk label should be facing up and toward you).

*//e users:* If you have more than one disk drive, be sure you are using Drive 1 (the drive whose light comes on when you turn on the Apple).

*//c users:* Always use the built-in disk drive with SwyftWare.

Close the disk drive door. Turn on your video monitor. Now turn on the power to your computer. The disk drive will clackety-clack and whirl. When it quiets down you will see the first screen of the tutorial.

The on-screen instructions will guide you from there. Because the tutorial disk is write-protected, nothing you type can harm it. So don't worry about mistakes.

### **Fresh Starts for the Tutorial**

The tutorial has no error messages. That's because we have one simple remedy for all problems: *Turn off the computer, wait a couple of seconds, and turn it on again.*

*There's no need to struggle with a mistake — forget it and start over.*

**CAUTION:** *When typing your own things, be sure to save them before you turn off the computer. Turning off the computer to get a fresh start works only when the text you see on the screen has been saved on a disk. The tutorial is saved on the tutorial disk, so it comes back when you turn the computer on again. If it wasn't saved, it would be gone forever. (How to save things is explained in the tutorial and in Part 13: The DISK Command.)*

---

## HOW TO USE SWYFTWARE

YOU control the power switch, so YOU control the computer. If you start to feel lost, all you need to do is turn off the computer, wait a second, then turn it on again. You'll be back on this page, ready to go. Now try typing your name:

Did your name come out in ALL CAPITALS? If it did, firmly press and release the CAPS LOCK key (at the lower left corner) until you can type in small letters.

---

### *First screen of the tutorial*

While the tutorial is self-explanatory, you can, if you like, read the brief guide which follows.

### *What's in the Tutorial*

In easy steps the tutorial illustrates how to type, edit, move the cursor, highlight, and use the commands. This is roughly the same order the information is presented in the manual, making it easy to use the tutorial and the manual together.

Each lesson is very brief (about two paragraphs), and usually includes an exercise. Look for the word "Example". This tells you that what follows is something for you to do. In general, don't do anything unless it's an example.

If you find yourself wondering why things work the way they do, just keep moving — the answer is probably around the corner.

There are about 60 of these short lessons. They are numbered. If you keep track of the number of your screen, it will help you get back in case you get lost. The tutorial explains how.

### *Bleeps While Doing the Tutorial*

A bleep is an audio warning signal from your Apple (*Part 23* covers them in detail). If you're doing the tutorial and the Apple starts to bleep every time you tap a key, it means that there is no more room to type anything. This very rarely happens, but since it is possible, here's what to do: Turn off the computer, wait a couple of seconds,

then turn the computer on again. The tutorial will start over with plenty of free space for typing.

**WARNING:** *The above technique applies to the tutorial only. We discuss the proper technique in Part 13, "When You Run Out of Room."*

Once the tutorial is on the screen, you can get back to the lesson you were working on by holding down the LEAP FORWARD (Solid-Apple) key and typing the number of the lesson. Make a mental note of that number before you turn off the computer.

### ***SwyftWare Not Running?***

A few very early Apple //e's (manufactured before March 1981) were shipped with incorrect ROMs and will not work with SwyftCard. If your //e is one of these, see your dealer about getting a replacement ROM.

A disk created on SwyftCard will not start up on an Apple //c or an Apple //e without SwyftCard unless the computer is already running SwyftWare. Follow these steps to convert a SwyftCard-created disk for use without a SwyftCard:

- You may want to first save whatever you were doing. Put a SwyftWare disk in the //c (or //e without SwyftCard) drive and load it (press CONTROL-OPEN-APPLE-RESET).
- Remove the SwyftDisk, put the SwyftCard-created disk in the drive and use the DISK command (USE FRONT L).
- Type CALL 3600, highlight this expression, and use the CALC command (USE FRONT G).
- Use the DISK command again. The disk will now run on either a //e or //c.

**WARNING:** *If you fail to use the DISK command after doing CALL 3600, you will lose your text.*

## **Part 6: Starting Something of Your Own**

### **Empty Text**

This is what your screen will look like when SwyftWare is ready to go, but nothing has been typed yet:



*flashing object*

*This is SwyftWare's version of a blank sheet of paper.*

We call this an *empty text*. An empty text is ready to receive your inspiration. How you get an empty text differs slightly, depending on whether you have SwyftCard or SwyftDisk.

### **Getting an Empty Text With Apple Ile and SwyftCard**

The SwyftCard must be installed in Slot 3 (*Part 3* describes installation).

Make sure there is no disk in the drive. Turn on your monitor and let it warm up a few seconds. Turn on your Apple. The drive will whirl and the screen will come to life. You'll see the copyright notice for a second or two while the drive runs.

SwyftWare 1131 Copyright 1986 Information Appliance Inc.

*Copyright notice. It appears for only a few moments.*

The next thing you see will be an empty text. You can type all you like now even though there is no disk in the drive, but you cannot save what you type without a disk. See *Part 13* for information on saving what you type.

## ***Getting an Empty Text With SwyftDisk***

Your SwyftWare package should have contained two diskettes. You will need the one labeled "SwyftDisk."

Turn off the power to your Apple //c or //e computer. Place the SwyftDisk disk in the drive. Turn on the monitor and let it warm up a few seconds. Now turn on the computer. The drive will whir and the screen will come to life. You'll see the copyright notice for a second or two while the drive runs (illustrated on the previous page).

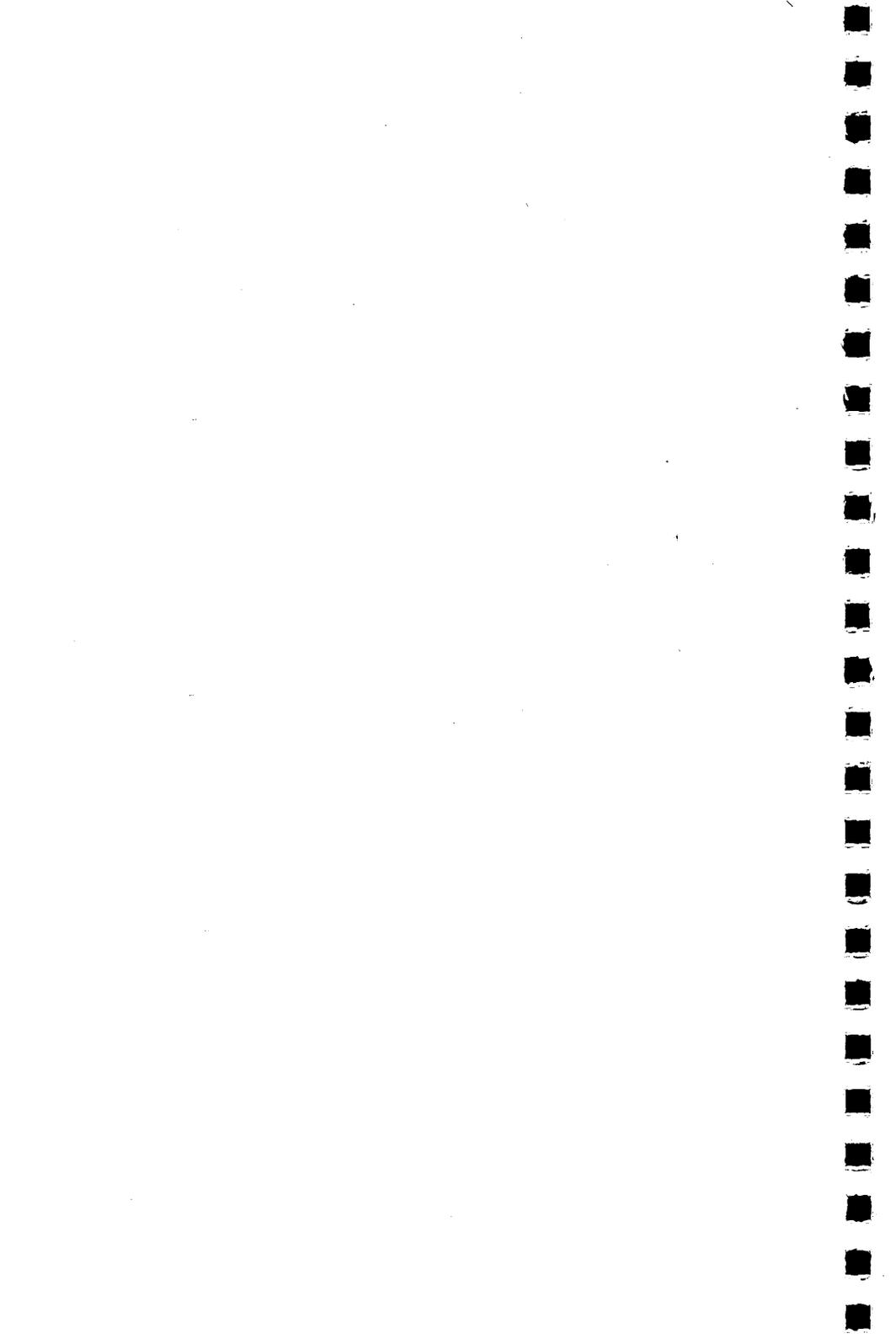
The next thing you see will be a empty text. You can type all you like now, but you cannot save what you type without putting a new, blank disk in the drive. See *Part 13* for information on saving what you type.

**Note:** See Part 13, "How to Delete the Entire Text," for another way to get a empty text with SwyftWare.

## **Chapter II**

### **Editing**

<i>Part 7</i>	<i>Typing</i>	<i>15</i>
<i>Part 8</i>	<i>Pagination</i>	<i>21</i>
<i>Part 9</i>	<i>Cursor Motion</i>	<i>24</i>
<i>Part 10</i>	<i>Extending the Highlight</i>	<i>32</i>
<i>Part 11</i>	<i>Delete</i>	<i>35</i>
<i>Part 12</i>	<i>The INSERT Command</i>	<i>37</i>



## **Part 7: Typing**

### **Characters and Text**

When you type, you create *characters* and *text*. *Characters* are individual letters, punctuation marks, symbols, and so forth. *Text* is what they add up to: all the characters you will be able to see on the screen. When we write *Text* with a capital "T," we mean the entire text, all the characters in the computer at one time.

### **The Cursor and the Highlight**

The *cursor* is the blinking object on the screen. It usually appears to the right of a solid, unblinking rectangle called the *highlight*. The cursor and highlight mark your place on the screen when you are typing.

### **What They Do**

The cursor shows you where the next character you type will appear. The highlight shows you what will be erased if you tap the DELETE key.

### **When You Can Type**

You can begin typing as soon as you see the blinking cursor on the screen.

```
===== 0 =====  
You can begin typing as soon as the blinking cursor appears on the screen (just  
imagine the small, shaded square at the end of this line blinking on and off, on  
and off). ■  
===== 1 =====
```

### **Where the Type Will Appear**

Your typing will appear on the screen at the position of the cursor. As you type, the cursor will move ahead, indicating at all times where your typing will next appear.

## ***How to Type Capital Letters***

Hold down a SHIFT key and, while you hold it down, tap the letter key. We say "tap" because if you do not quickly release the letter key, you will get many letters. That is why in writing this manual we have been careful to use the word "tap" when we mean a brief tap, and "hold" when you need to hold the key down.

## ***When There Are Two Symbols on One Key***

To type the lower symbol, simply tap the key. To type the upper symbol, hold down one of the SHIFT keys and tap the key of that symbol. For example, to type an asterisk (\*), hold down a SHIFT key, and, while holding it, tap the "8" key.

***Note:*** The keys labeled *ESC/PAGE*, *TAB/LEAP AGAIN*, *CONTROL/USE FRONT* are exceptions. Holding *SHIFT* will not affect these keys. Their functions are discussed elsewhere.

## ***The Labeled Keys***

The letter keys we asked you to label do not type any differently than the non-labeled keys. Holding a SHIFT key and typing a "D", for example, results in a capital D, not a "SEND". The labeled keys will be explained as we go.

## ***Word Wrap***

When a line fills up, the computer automatically begins a new one without ever breaking a word. This is called *word wrap*. Because of word wrap, you do not have to tap the RETURN key at the end of lines. This saves a lot of thought and effort compared to a typewriter. Word wrap also works in reverse. If you remove part of what you have already typed, the material ahead of the part you took out will move back and fill in the gap.

## ***Spaces in Word Wrap***

Typing a space or two at the very end of a line will trigger word wrap if they come after a word. The spaces and the word preceding them move down to the beginning of the next line. This is done to keep the left edge of the text even. If the space alone was moved down, the left edge would look ragged.

## ***Typing in the Middle of Something***

What you type is always inserted, it never replaces what is already there. The letters to the right of the blinking cursor and at the cursor itself are pushed out of the way when you type. Line ends adjust automatically.

## ***Cast of Characters***

This is a complete list of the 97 characters available in SwyftWare:

*Lowercase letters:*    a b c d e f g h i j k l m n o p q r s t u v w x y z

*Uppercase letters:*    A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

*Numerals:*            0 1 2 3 4 5 6 7 8 9

*Punctuation:*        , . ; : ' " - ? !

*Special symbols:*    @ # \$ % ^ & \* ( ) \_ = + [ ] { } \ | < > / ' ~

*Format characters:*    space, return, page

## ***Format Characters***

The following three characters, called *format characters*, determine the spatial arrangement of text on the screen and in the printout. As far as SwyftWare is concerned, format characters are no different than letters, punctuation, and numerals.

***Space:*** An electronic separator the width of one letter. It is produced by the SPACE BAR and separates words on screen.

```
===== 0 =====
Spaces are the electronic bricks that hold the words apart. Sometimes they hold
the words quite far apart.
===== 1 =====
```

***Return:*** Ends the line you are typing and moves the cursor to the beginning of the next line. Invisible except when included in the highlight, where it appears as a shaded spot.

```

===== 0 =====
←spaces → Return characters end lines. ←return
|return
And, return characters create blank lines between paragraphs. ←return
|return
===== 1 =====

```

*Returns are usually invisible characters.*

**Page:** Produced by the PAGE key (marked ESC on the Apple keyboard). Forces the ending of a page (see *Part 8*, "Page Characters").

```

===== 0 =====
end of page 1 ↓
===== 1 =====

```

*Page characters zero and one*

### **Underlining**

Unfortunately, neither the Apple //e nor //c can show underlined text on the screen. Thus we have to resort to a trick that allows you to get underlined text from your printer and at the same time see which text is underlined on the screen.

The trick is to type an underline character *instead* of a space in front of the word or words you want to underline, then type another underline character *instead* of a space after the word or words you want to underline. Everything between the two underline characters will be underlined in the printout; the underline characters themselves will print as spaces.

This means that you cannot underline a single character in the middle of a word. Underlining a word at the beginning of a line is no problem: make the underline character the last character of the preceding line.

To underline the last word of a sentence, type the underline character *instead* of the first space after the period. The period will be underlined, but this is generally acceptable.

To print a continuous line (with most printers), as when creating a form, type as many underline characters in a row as necessary. The

first and last will become spaces, but the rest will print as underline characters.

An isolated underline character can be obtained by typing three underline characters in a row.

Two return characters in a row will turn off underlining. This stops your whole text from being underlined in case you forget to turn the underlining off. A page character will also turn off underlining.

***Underlining:  
Screen appearance***

Underlining a single word.

Fill in the blank: \_\_\_\_\_

Underlining an entire line..

Underlining is stopped by two  
consecutive returns.

***Underlining:  
Printout appearance***

Underlining a single word.

Fill in the blank: \_\_\_\_\_

Underlining an entire line.

Underlining is stopped by two  
consecutive returns.

We feel that this underlining technique is not truly satisfactory, but it is a compromise between the design limitations of the Apple //e and //c and our desire to keep the arrangement of text on the screen looking the way it will look on the printer. If the underline characters disappeared when printed, rather than being replaced by spaces, then items that lined up on the display would no longer line up in the printout. (Underlining may not work with your printer until you read *Part 17* and make the appropriate settings.)

### ***Other Features of the Keyboard***

***DELETE key*** Erases material from the screen (see *Part 11: DELETE*). You can always tell what will be erased when you press DELETE — it's whatever is in the highlight (the solid unblinking rectangle).

***CAPS LOCK*** An alternate-action switch that, when engaged, switches all letter keys to capitals. Numerals, symbols, and punctuation are not affected. (This key, if left depressed, may

interfere with cursor movement. See *Part 9, "Capital Letters in Leap Patterns."*) Remember this homily: If CAPS LOCK is left depressed, you soon will be too.

**TAB** Acts as if you had pressed the SPACE BAR enough times to reach the nearest preset tab stop to the right of the cursor. Tab stops are at columns 5, 10, 15, 20, ..., 75. Pressing TAB adds space characters to the text; it does not just move the cursor across the screen.

**USE FRONT** Press and *hold* USE FRONT (CONTROL on the Apple keyboard) when you want to use the actions shown by the labels on the *fronts* of the letter keys. Note that you cannot type any letters or move the cursor while USE FRONT is held down. The influence of the USE FRONT key ends as soon as it is released.

### **Keys You Can Use While USE FRONT Is Held Down**

<b>Key</b>	<b>Label</b>
A	INSERT
D	SEND
G	CALC
L	DISK
N	PRINT
Z	No label (see <i>Part 20, "Stopping Incoming Text and Restarting It"</i> )

**Autorepeat** If you press and hold DELETE or any character-producing key for more than half a second, it will repeat at a rate of about 15 characters per second. Keys that do not autorepeat include USE FRONT, SHIFT, CAPS LOCK, and the LEAP keys (Open-Apple and Solid-Apple).

**Arrow keys** Located in the lower right-hand corner of the keyboard, these are the cursor control keys for the Apple system. They are disengaged while you are in the SwyftWare system. They become operational again when you use Apple software. We do not use them in SwyftWare because leaping is a much faster, more efficient way of moving the cursor.

## **Part 8: Pagination**

### **Page Breaks and Page Numbering**

Unless adjusted otherwise, SwyftWare allows for 54 printed lines per page. When you type enough lines to fill a printed page, SwyftWare automatically starts a new page and — by displaying a single-dashed line with the page number at center — shows you on the screen where the paper page produced by the printer would end. The number will print at the bottom center of the page, but the dashed line does not print. Page numbers 0 and 1 will not normally print (this can be changed). Page numbering can be changed (see *Part 18: Formatting the Printout*).

### **Page Characters**

You can force a page to end and a new page to begin at any point in the text by tapping the PAGE key (marked ESC on the Apple keyboard). This inserts a page character into the text. When sent to the printer, the page character causes the printer to advance the paper, print the page number at the bottom center of the page, and then advance to the top of a new page.

The on-screen symbol of the page character is a double-dashed bar with a number at center. Both the double-dashed bar of a page character and the single-dashed line of a page break may be thought of as representing a perforation in a physical page of paper.

----- 25 -----  
*page break*

===== 26 =====  
*page character*

*A page break is inserted when you type enough lines to fill a printed page. A page character will force a page to end at any point you desire.*

### **Uses of Page Characters**

Page characters are generally used to begin and end a document. For example, when you begin your letter to Aunt Ida, type a page character to ensure that "Dear Auntie" (or whatever you type next) will

be at the top of a new page (a top margin will automatically be inserted in the printout).

When you finish writing the letter, type a page character to close the page you are working on. Pagination between the page characters is automatic, that is, page breaks will appear automatically as you type along and fill the pages.

A page character also ensures that the next thing you type will remain at the top of a new page regardless of changes you may make in the text that precedes the page character.

Another use of a page character is to force a page to have less text on it. For example, if you wanted to leave half a page blank for an illustration, you would type half a page of text and then type a page character.

A page character can also eliminate an inconvenient page break. To take out a page break that comes just after the first line of a paragraph, for example, simply type a page character immediately above the split paragraph. This moves the whole paragraph to the top of the next page.

### ***Differences Between Page Breaks and Page Characters***

Unlike page characters, page breaks are not characters in the text, just indications of where a new page will automatically start because the preceding page was filled. You might think of them as mileage markers — signs of going a certain distance, or filling a certain amount of text.

Because page breaks are not characters, you cannot erase them. Also, because the SwyftWare cursor moves only to characters in the text, you can move the cursor to a page character, but not to a page break (cursor motion will be fully explained in *Part 9*).

**Note:** *The first and last page characters — numbered 0 and 1 in an empty text — are exceptions: though you can move the cursor to them, you cannot erase them.*

If you insert new text above a page character, the page character moves out of the way with the rest of the displaced text. If you insert new text above a page break, the page break stays where it is while

the text that is pushed out of the way flows around it. (Remember that it doesn't cost anything to take a bit of time and try these features out on your own system.)

Unlike page characters, page breaks do not turn off underlining. This is because page breaks pop up in your text wherever they are needed — and one might just pop up in the middle of an underlined phrase.

### ***Maximum Number of Pages***

SwyftWare permits no more than 200 page breaks or page characters per Text. In practice, this 200-page maximum is rarely even approached, as SwyftWare's storage capacity permits between 15 and 20 densely typed pages per disk.

This does not mean, however, that you cannot have page numbers higher than 200. The page numbering of a Text can be adjusted (see *Part 18, "Changing Page Numbers"*). A single document may span several disks: pages 1-20 on Disk A, pages 21-40 on Disk B, pages 41-60 on Disk C, and so on.

## Part 9: Cursor Motion

### Leaping

SwiftWare's cursor — the flashing object on your screen — is capable of moving anywhere in the text almost instantly. The term *leap* has been adopted to describe this new mode of travel. A leap is extremely fast, averaging less than half a second no matter what the distance.

### Creeping

The cursor may be moved short distances by creeping. *Creeping* moves the cursor character-by-character. Though usually much slower than leaping, creeping is often convenient for very short moves.

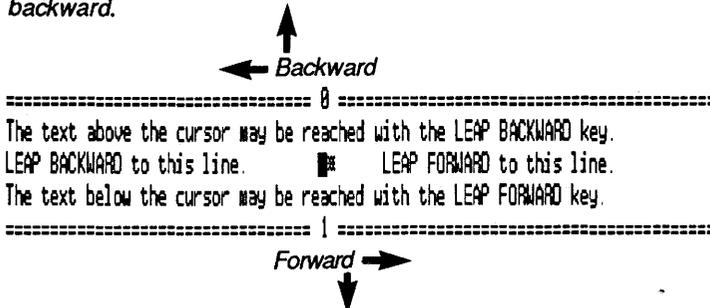
### The LEAP Keys

The Open-Apple key (left side of SPACE BAR) and Solid-Apple key (right side of SPACE BAR) control the movement of the cursor. Open-Apple is called *LEAP BACKWARD*, Solid-Apple is called *LEAP FORWARD*.

Pressing and *holding* LEAP BACKWARD readies the cursor for a leap backward to any character in the text. Pressing and *holding* LEAP FORWARD readies the cursor to leap forward to any character in the text.

### Forward and Backward — Where They Are in Relation to Cursor

The line of text to the right of the cursor and all lines below the cursor are *forward*. The line of text to the left of the cursor and all lines above it are *backward*.



*Forward and backward in relation to the cursor*

## ***How to Creep***

To creep forward, tap repeatedly on the LEAP FORWARD key. Each tap advances the cursor one more character.

To creep backward, tap repeatedly on the LEAP BACKWARD key. Each tap moves the cursor backward one more character.

Creeping does not autorepeat. If you find yourself wishing that creeping would autorepeat, this only means that you are creeping when you should be leaping.

## ***How to Leap***

First choose the exact character you want to leap to. This is called the *target character*. Any character among the cast of 97 (see Part 7, "Cast of Characters" ) may be used as a target, including a space, a return, or a page character.

Hold down the appropriate LEAP key, and, *while you hold it down*, tap the key of the target character. The cursor will leap in the direction indicated by the LEAP key to the nearest instance of the target character.

## ***The Importance of Letting Go***

After the cursor reaches the target character, let go of the LEAP key. You cannot type text or start a new leap until you let go.

Also, *don't let go until you are satisfied with where the cursor has landed*. Until you let go, you have the option of undoing the leap, that is, leaping back to where you were (see "Undoing a Leap," this chapter).

## ***Patterns — How to Reach Any Target in One Leap***

*Patterns* are the characters (or single character) you type while holding down a LEAP key. A pattern may be up to 31 characters in length, much longer than you are likely to need. In practice, most patterns are three or four characters at most.

*Patterns permit you to leap directly to a target.*

The first character in the pattern is the target character. The characters that follow establish exactly which instance of the target character you want.

To use a pattern to leap somewhere, hold down the appropriate LEAP key, type the target character, then continue to type the characters that follow the target character in the text, just as if you were typing that part of text all over again. Stop typing and let go of the LEAP key when the cursor reaches the target.

For example, if you type "ab" while holding down LEAP FORWARD, the cursor will leap ahead to the nearest instance of an "a" followed by a "b". If you now type an "o" (without letting go of the LEAP key), the pattern becomes "abo". The cursor will leap in the same direction to the nearest instance of an "abo" pattern in the text (unless the cursor was already on the "a" of "abo", in which case it won't move).

Note that no matter how lengthy the pattern becomes, the cursor still leaps to the first character. If you typed the pattern "about", the cursor would leap to that nearest "a" followed by the letters "bout".

### ***Nonexistent Patterns***

If the pattern cannot be found, the cursor will return to its starting place. This may happen so fast that it looks as if it hasn't moved at all.

### ***The Most Common Mistake In Leaping***

Trying to leap again by tapping the same key again (and again) is the most common mistake in leaping. If you want to leap again to the same character, type that character once, and, while you continue to hold down a LEAP key, tap the LEAP AGAIN (TAB) key.

Tapping again and again on the same character key in an attempt to repeat a leap usually results in a nonexistent pattern ("yyyyyyy", for example). The cursor leaps back to where it started from and stays there. You wonder why it isn't moving, even though you are tapping and tapping.

It's because you're making *the most common mistake in leaping*.

### ***Undoing a Leap***

To undo a leap, continue to hold the LEAP key and change the pattern so that it won't match anything in text. Adding x's to the pattern is one way. Another way is to tap a bunch of keys at once. The cursor will return to its starting place because the nonexistent pattern cannot be found.

Again, the importance of *not* letting go: You must undo the leap *before* you let go of the LEAP key.

And again, the importance of *letting* go: You must *let go* of the LEAP key *after* the leap is undone. Only then can you type or begin a new leap.

### ***Correcting Errors in a Pattern***

As long as a LEAP key is held down, characters can be deleted from a pattern. One press of DELETE removes one character from the pattern. New characters can be typed to replace those that have been deleted.

*The LEAP key must be held down continuously while deleting and correcting a pattern.*

As you erase characters, the pattern grows shorter. Consequently, the cursor retraces its steps. If you delete all the characters in the pattern, the cursor will return to the point where it stood when you first pressed and held the LEAP key. This is a way of fixing a typing error you make while leaping, and it is also another way you can undo a leap.

### ***LEAP AGAIN***

Holding down a LEAP key changes the TAB key to the *LEAP AGAIN* key. While holding a LEAP key, each tap on the LEAP AGAIN key causes the cursor to leap again to the same pattern. This way you don't have to retype a pattern when you want to leap to it again.

As soon as you release the LEAP key, the LEAP AGAIN key goes back to being a TAB key.

If you tap LEAP AGAIN repeatedly, you will eventually find every instance of a pattern in the text. When you have found every instance of a pattern, the cursor will return to where it was when you first held down the LEAP key.

You can use LEAP AGAIN to circle the text continuously in the same direction, as if circling a racetrack. You know you've made a complete tour of the text when the cursor returns to its starting point — the place it was when you first held down the LEAP key.

LEAP AGAIN autorepeats. Pressing and holding LEAP AGAIN for more than half a second causes the cursor to move rapidly from one instance of a pattern to the next.

## ***Pattern Memory***

Once a pattern is used, it is retained until you press and hold down a LEAP key and type another pattern.

### ***To Use the Same Pattern to Leap the Cursor in Either Direction***

Let go of the LEAP key you are using. Press and hold down the other LEAP key. Tap the LEAP AGAIN key. Because the pattern is saved, you can leap again in either direction by using the LEAP AGAIN key.

### ***To Reuse the Same Pattern***

Hold down either LEAP key and tap LEAP AGAIN. The last pattern used in leaping is still available after you have typed something or used the DELETE key to delete character-by-character.

### ***Capital Letters in Leap Patterns***

***— or, a handy thing, so long as the CAPS LOCK key is not stuck down.***

Lowercase letters in a leap pattern match both uppercase and lowercase letters in the text. This means that you can look for an instance of a word without having to worry if it is capitalized. But uppercase letters in a leap pattern match *only* uppercase letters in the text. By using a capital letter in a leap pattern, you can look for, say, the word "Leap" at the beginning of a sentence, ignoring all the words "leap" in the middle of the sentence.

This explains why leaping seems to be out of order if the CAPS LOCK key is left in its down position: it CAPITALIZES all your letter patterns, and CAPITAL LETTER PATTERNS will not match lowercase letter patterns.

The following chart shows what SwyftWare will find when you use capital letters in a leap pattern.

<b><i>Leap Pattern</i></b>	<b><i>What It Will Find</i></b>
me	me, Me, mE, ME
Me	Me, ME
ME	ME

## ***Leaping Word-by-Word***

Words are usually separated by spaces. To leap from word to word, leap to a space (hold down a LEAP key and tap the SPACE BAR), then use the LEAP AGAIN key. Press and hold the LEAP AGAIN key to engage autorepeat if you wish to move quickly from word to word (though still space-by-space).

## ***Leaping Sentence-by-Sentence***

Sentences generally close with a period. To leap from sentence to sentence, leap to a period (hold down a LEAP key and tap the period key), then use the LEAP AGAIN key.

Another approach: Most typists separate sentences with two spaces, so you can use two spaces as a pattern for leaping from sentence to sentence. This way you don't leap over sentences that end with a question mark or exclamation point. Hold down a LEAP key and tap the SPACE BAR twice, then use the LEAP AGAIN key.

## ***Leaping From Paragraph to Paragraph — Scrolling***

To *scroll* means to move the text smoothly and continuously past the window of your screen, as if it were a continuous roll of electronic paper parading by. Leaping from paragraph to paragraph provides — but with greater speed — the same function as scrolling does in other systems.

Paragraphs are generally separated by two returns, thus we use a pattern of two returns to leap from paragraph to paragraph.

***How to scroll:*** Hold down a LEAP key, tap the RETURN key twice, then tap the LEAP AGAIN key (while you hold the LEAP key). To scroll more quickly, press and hold LEAP AGAIN. This engages autorepeat. To change directions, just let go of the LEAP key you are using, press and hold the other LEAP key, then tap or press the LEAP AGAIN key.

If your text has only one return between paragraphs, leap to a single return, then use the LEAP AGAIN key to scroll. In the unlikely event that your text has no returns at all, use another common character, such as a period, for your scrolling pattern.

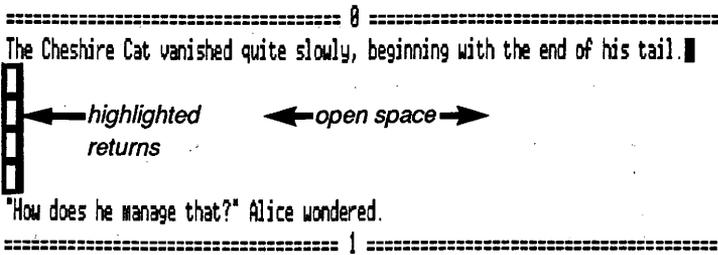
## ***Leaping From Document to Document***

Documents generally begin with a page character. To leap from document to document, hold down a LEAP key, tap the PAGE key, then use the LEAP AGAIN key.

## ***Leaping to Open Space on the Screen***

*Open space* means blank parts of the screen, such as beyond the ends of lines or between blocks of text.

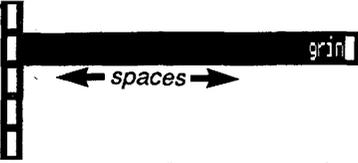
Open space is created by a return character ending a line or by return characters along the left margin. You can see the returns if they are in the highlight (*Part 10* describes how to make the highlight cover more than one character).



*Open space between lines of text. The returns are highlighted.*

The open space to the right of the returns has no characters in it, therefore you cannot leap to it. If you wish to move the cursor into the open space, leap to a return next to the desired line. Use the TAB key or the SPACE BAR to type enough spaces to move the cursor to the middle of the screen.

=====  
The Cheshire Cat vanished quite slowly, beginning with the end of his tail, and ending with his grin, which floated in the center of Alice's video monitor. █



I think I see now," Alice said, as she highlighted the text.

=====  
|

*Text that appears in the middle of open space is held there by space characters and returns.*

**Note:** *The space characters and returns in the screen illustration above are highlighted so you can see the structure of the text.*

## ***Part 10: Extending the Highlight***

The highlight — the bright, solid rectangle that accompanies the flashing cursor on the screen — can be extended to include two or more characters at once. (The highlight normally covers only one character.)

A contiguous portion of the text that includes two or more characters is called a *chunk*. A highlighted chunk of text can be manipulated in a variety of ways by using SwyftWare's commands.

### ***To Highlight a Chunk of Text (Two or More Characters)***

Leap the cursor from one end of the desired text to the other, then press both LEAP keys at once. You do not have to let go of one key before you press the other. (After letting go of both LEAP keys you can still highlight the text covered in your last leap until you type something or you leap again.)

A highlighted chunk of text remains highlighted until you type something, use the DELETE key, or press a LEAP key.

You can leap from the first character of the chunk to the last, or from the last to the first — the direction you leap is unimportant.

### ***Cover the Chunk in One Leap***

When you let go of the LEAP key, the boundary of the highlight is set. Therefore, use only one leap (one pattern) to cover the text you wish to highlight. It's OK to use LEAP AGAIN while leaping across the chunk of text because LEAP AGAIN does not force you to let go of the LEAP key.

### ***The Exact Boundaries of an Extended Highlight***

The highlight will extend from the character that was inside the highlight at the beginning of the leap to the target character at the end of the leap.

### ***Highlighting Recently Typed Text***

If you've typed some text since the last leap or creep, and you press

both LEAP keys at once, the highlight will extend over the text you've just typed.

### ***How to Undo an Extended Highlight***

Tap either of the LEAP keys. The highlight will collapse back to the size of a single character.

Tapping LEAP BACKWARD will collapse the cursor and highlight on the first character of the text that was highlighted.

Tapping LEAP FORWARD will collapse the cursor and highlight on the last character of the text that was highlighted.

In practice, it is usually unnecessary to undo the highlight — all you have to do is begin typing. The highlight will collapse and the new characters will appear on the screen at the position of the cursor.

### ***Rehighlighting***

If you undo a highlight by tapping a LEAP key and then press both LEAP keys at once, the highlight will reappear exactly as before. As soon as you type or leap again, you lose the option to rehighlight.

After you invoke the PRINT, SEND, or CALC commands, you can rehighlight the text simply by pressing both LEAP keys at once.

### ***Adjusting the Boundaries of the Highlight by Creeping***

Sometimes you find that you have highlighted a chunk of text that is off at one or both ends by just a little bit. This can happen due to a leaping error, but usually it is because some other target was more convenient than the exact end of the chunk you wanted.

To adjust one end or the other of the highlight, collapse the highlight to the left or right end (by pressing and releasing the left or right LEAP key) and creep the cursor to the right or left until it is at the right place. Then press both LEAP keys to rehighlight the chunk you wanted.

### ***Autoextending Highlight***

Three commands — PRINT, CALC, and SEND — permit you to highlight automatically, without pressing both LEAP keys at once beforehand.

If you use any of these three commands after a leap, the text you covered in the last leap will automatically highlight and the command will be executed.

If you use any of them after typing something, the highlight will extend to everything you've typed since the last leap and the command will be executed. This can be very handy, especially with SEND.

To find out more about these commands, see *Parts 14, 17, and 20*.

### ***Typing and Highlighting a Chunk***

If you press both LEAP keys at once, then type some text, and then again press both LEAP keys at once (or allow the PRINT, CALC, or SEND commands to automatically highlight for you), the text you just typed will be highlighted.

Here's a practical example — how to write yourself a quick reminder: Press both LEAP keys at once (so that typing will start a brand new chunk), then type, "Remember to buy two quarts of milk!" Then use the PRINT command (which automatically highlights what you just typed), then press both LEAP keys at once (to rehighlight what you typed), and finally tap the DELETE key to erase the message.

## **Part 11: DELETE**

### **To Delete Single Characters**

Tap the DELETE key. DELETE erases whatever character is inside the highlight. Always. If the character you want to erase is already in the highlight, just tap DELETE. If not, move the cursor to the character you want to erase and tap DELETE.

DELETE is an autorepeating key. If you want to erase a string of characters rapidly, hold down DELETE (with or without autorepeat, this process is called *character-by-character deletion*).

Use character-by-character deletion with care — once deleted, these characters are gone forever.

### **Wide Cursor and Narrow Cursor**

When the cursor and the highlight are both on the same character we call the cursor *narrow*. The cursor becomes narrow whenever you press a LEAP key, leap, or creep somewhere.

As soon as you begin to type, or a command inserts characters into the text, the cursor and the highlight split apart. We call a cursor *wide* when the flashing cursor and the highlighted character are next to one another.

### **Backward Delete**

A wide cursor deletes to the left. This is called *backward delete*. Backward delete resembles backspacing on a typewriter.

### **Forward Delete**

A narrow cursor deletes to the right. This is called *forward delete*. The cursor does not move when you forward delete. It stands still while vacuuming up the characters to its right.

### **Why Two Delete Directions?**

Now that we've told you about forward and backward delete, you should feel free to forget all about them most of the time. That's because DELETE generally does what you want it to do without any planning.

When you make a mistake while typing, the error is to the *left* of the cursor and DELETE backspaces to the left. When you LEAP to something you want to delete, you naturally leap to its *beginning*, putting the text you want to delete on the right side of the cursor. And that's the way DELETE works after you leap — to the right, automatically.

### ***Delete Turnaround***

Character-by-character delete changes directions when you come to the first or last page breaks. For example, if you backward delete (wide cursor) up to the first page break (page break zero), the cursor will switch to narrow and begin to delete forward.

### ***To Delete Two or More Characters at One Keystroke***

Highlight the text, then tap the DELETE key. The highlighted text will vanish and the remaining text will close in to fill the gap.

### ***To Bring Back the Deleted Chunk***

Use the INSERT command: Press and hold USE FRONT, and, while you hold down USE FRONT, tap the "A" key.

Unlike a deleted chunk, text that has been deleted character-by-character cannot be retrieved.

### ***How Much You Can Delete***

As much as you have. There is no other limit on the size of a delete.

## ***Part 12: The INSERT Command***

### ***How to Insert New Text by Typing***

Move the cursor to the place you want to add material and just begin typing. The character on which you place the flashing cursor will be pushed ahead of the new text.

### ***How to Move a Chunk of Text Using the INSERT Command***

Extend the highlight and DELETE the chunk. Leap the cursor to the new location, then use the INSERT command (press USE FRONT A). The deleted chunk of text will reappear at the new position of the cursor. The moved chunk is left highlighted to make it easy to move again.

### ***To Make a Single Copy of a Chunk of Text***

Extend the highlight and DELETE the chunk. Then use the INSERT command (press USE FRONT A). A DELETE immediately followed by an INSERT leaves the original copy exactly in place. Then leap to the location in the text where you want the copy to go. Use the INSERT command to insert the copy.

### ***To Make Multiple Copies of a Chunk of Text***

After deleting the chunk, use the INSERT command repeatedly. You can place the multiple copies where the original was located, or (by leaping before each INSERT) at desired locations in the text. If you hold USE FRONT A down for more than half a second, autorepeat takes over (autorepeat is slower for the INSERT command; expect a short delay between copies).

### ***Lifetime of the Deleted Chunk***

The deleted text remains in memory, available for INSERT, until you highlight and delete another chunk. Only the most recently deleted chunk is available (and taking up space). Each time you delete a chunk, the previously deleted chunk is gone forever. If you save the text on a disk, the most recently deleted chunk is also saved.

## ***Search and Replace***

It sometimes happens that you want to replace some or all occurrences of a word with some other word. There is a relatively easy way to do this. Say that you want to change "Smith" to "Jones" wherever it occurs in a letter:

1. Start by highlighting and deleting the replacement word as one chunk (character-by-character deletion will not do): Type "Jones" once, leap backward to "J", highlight the word by pressing both LEAP keys, then press the DELETE key.
2. Now leap to the word you want to replace and delete it character-by-character: Leap to "Smith" and tap the DELETE key five times.
3. Use the INSERT command: Press USE FRONT A and "Jones" pops out on-screen.

The rest is very easy. Just

4. LEAP AGAIN
5. Let go and tap the DELETE key five times.
6. Use the INSERT command.

Repeat steps 4, 5, and 6 for each instance of "Smith" you want to change. If there is an instance you want to leave alone, just LEAP AGAIN past it to the next instance.

Like a lot of what you can do with SwyftWare, it is easier to do it than to read the explanation. Try it once or twice.

## ***Chapter III***

### ***Saving and Calculating***

<i>Part 13</i>	<i>The DISK Command</i>	<i>39</i>
<i>Part 14</i>	<i>The CALC Command</i>	<i>51</i>
<i>Part 15</i>	<i>CALC Command Miscellany</i>	<i>53</i>
<i>Part 16</i>	<i>BASIC in the Swyft Environment</i>	<i>55</i>



## **Part 13: The DISK Command**

### ***The Golden Disk Rule***

Everything will work fine if you follow this simple rule: Always use the DISK command (USE FRONT L) just before you take a disk out; always use the DISK command just after you put a disk in.

The manuals that came with your Apple computer explain the mundane rules about taking care of disks, which we paraphrase as follows:

*Don't use them as coasters,  
Don't put them in toasters,  
Don't leave them near magnets,  
Don't tear them to fragments.*

### ***SwiftWare Needs Only One Disk Drive***

Most products require two disk drives, either to facilitate copying or so that the system can reside on one drive and your data on another. SwiftWare does not have these problems (copying takes only a few seconds, and the system is always in memory). If you have more than one disk drive, the DISK command will use Drive 1. If you're not sure which drive is Drive 1, open the disk drive doors, use the DISK command (hold down the USE FRONT key, and, while you hold it, tap the L key), and see which red light comes on.

### ***The Text***

As we have said, text (little "t") is the set of characters that you're able to see on the screen. The Text (big "T") is all the text in the computer at one time, the text taken as a whole. Each Text has room for over 40,000 characters. That translates to roughly 35 double-spaced pages of average density, or 14 pages of really dense single-spaced text. You can store one Text on a disk. To put it another way, you can have as many Texts as you have disks.

### ***Writing on Sand in the Windblown Desert***

Such is the nature of your on-screen text. It can be fun to see your ideas appear on the screen, and to edit and print them. But — unless

you use the DISK command — that text is lost as soon as you turn off your Apple, and nothing can bring it back. A disk provides relatively safe and permanent storage. Thus you need the DISK command to store information on disks while the Apple's power is off, and to extend the amount of text that you can work with.

### ***Storing Information Between Sessions***

Just before you turn off your Apple, use the DISK command to save your work. After the save, the cursor blinks at a faster rate (the "happy" cursor). If you see a happy cursor, you know your text is safe.

When you save a Text for the first time, you should have a brand-new, blank disk in the drive. You can get a blank disk fresh out of the box, or, using a technique described later in this chapter, you can "blankify" one. If the disk in the drive is not a blank, SwyftWare will not save on it. Instead, it will give you an audible warning signal — a "bleep" — and do nothing. This protects the Text on your disk from accidental ruin, since you wipe out what is on the disk whenever you save what is on the screen.

SwyftWare also recognizes which Text goes with which disk, so don't change disks before you use the DISK command. If the disk in the drive is different than the one you started with, you will get a bleep. Again, this is a safety feature: saving your text on the wrong disk would destroy the text on the disk.

After you save, you can safely turn off the computer. Leave the disk in the drive if you like. When you come back and turn on your Apple again, the screen will come back just as though you had never turned it off. The cursor will be in exactly the same place; even the last chunk you erased will still be in storage, ready to be recalled with the INSERT command.

### ***What If the Power Goes Off by Accident?***

Suppose your pet rhino trips over the power cord to your Apple, or a persevering woodpecker finally topples the power pole down the street. Out goes the electricity to your computer, and with it goes all the work you've done since the last time you used the DISK command and saved your Text. Thus it's good to cultivate the habit of using DISK every few minutes: when you stop to think, or when you stand up to stretch your legs — whenever it won't interfere with the flow of

your work. It takes only a few seconds.

### ***Switching From One Text to Another***

Suppose you have one disk, the *phones* disk, with a list of phone numbers and addresses, and another disk, the *letters* disk, with correspondence on it. You arrive in the morning, put the *phones* disk in the drive, and turn on your Apple.

The phone numbers and addresses appear on your display and you add a new name to the list. Now you want to look at the *letters* disk.

Here's what you do: Use the DISK command. This will save the new name on the *phones* disk. Take out the *phones* disk and put in the *letters* disk. Use the DISK command again. After a few seconds your correspondence will show on the display.

### ***The DISK Command Protects Your Work***

What if you forgot to use the DISK command before taking the *phones* disk out of the drive? You put the *letters* disk in the drive and use the DISK command. This puts SwyftWare into a quandary. If it brings in the Text from your *letters* disk, it will wipe out and lose the new name and address you typed in. If it saves the text from your *phones* disk onto the disk in the drive, it will wipe out and lose your correspondence.

So instead of acting, SwyftWare sounds a bleep and does nothing. This reminds you to put the *phones* disk back in the drive, use the DISK command (which saves the new phone number and address), and then switch the disks.

After you've done all that and you use the DISK command again, SwyftWare thinks, "Aha! All the work on the *phones* disk is saved, so it is safe to bring in the Text from the *letters* disk." And that's what it does.

### ***Protecting the Contents of a Disk***

By putting a patch of adhesive-backed material called a *write-protect tab* over the notch in the side of a disk, you can protect the disk's contents from being changed. A supply of write-protect tabs usually comes with a new box of disks.

Write-protection is especially useful for disks you want to refer to without the risk of changing them accidentally. The tutorial, for example, is write-protected.

### ***Other Bleeps When Using the DISK Command***

If you use the DISK command to try to save your text onto a write-protected disk, SwyftWare will bleep to let you know that it can't do what you want. The fix? Take off the write-protect tab (if you really want to change the disk), or find another disk on which to save your Text.

There are two other reasons SwyftWare may bleep when you use DISK: an empty disk drive, or an open disk drive door. What to do? Put in a disk and close the door. Use the DISK command again.

### ***Summary of What to Do If You Get a Bleep When You Use the DISK Command***

1. Is there a disk in the drive? If not, put one in and close the door, try DISK again.
2. Is the disk drive door open? If it is, close it, and try DISK again.
3. Have you switched disks (e.g. gone from *phones* to *letters*) without saving any changes you made first? Switch back to the first disk and use the DISK command, then go ahead.
4. Have you switched disks and you don't want to save any changes you made in the text of the first disk? Delete the entire Text so that ONLY the two page characters 0 and 1 are left on the display (empty text), then use the DISK command. SwyftWare will bring the second disk on-screen without a bleep.
5. Is a large chunk of text highlighted? If a chunk of text is highlighted and you try to load another Text, SwyftWare will try to insert a copy of this highlighted chunk into the Text you are loading. If the inserted material won't fit, you get a bleep (see "*Moving a Chunk From Disk to Disk — Copy Up*," in this chapter). Tap a LEAP key to collapse the highlight to a single character, then try the DISK command again. Alternatively, highlight a smaller chunk and try the DISK command again.

## **How To Delete the Entire Text**

Leap to the first page character, number zero. Let go of all keys. Hold LEAP BACKWARD (the left LEAP key) and tap the PAGE key (ESC) once. This leaps you to the last page character in the text. Now press both LEAP keys at once. The entire Text highlights. Press the DELETE key. Done. The text now appears to be empty, but if you change your mind, you can retrieve the deleted material by using the INSERT command.

**A second method available only on SwyftCard:** Press and hold USE FRONT and the Solid-Apple key (right LEAP key). While holding both of these keys, press and release the RESET button. The copyright message will appear briefly, followed by a completely empty Text (page breaks 0 and 1 only). Let go of USE FRONT and Solid-Apple.

**CAUTION:** CONTROL-SOLID-APPLE-RESET completely wipes out the Text in the computer (not on the disk, of course). If you use the INSERT command now, you get a bleep. That is because there is nothing to insert — the Text is completely gone.

### **Force Loading —**

#### **Or, What to Do When Your Kid Bangs on the Keyboard and You Want to Go Back to the Way It Was**

To discard the Text on-screen and reload the Text on the disk, delete the entire text as described above, then use the DISK command. Don't worry, SwyftWare will not do a save, it will load the disk.

**CAUTION:** It is very important that you delete the entire text, leaving not a single character. If you use the DISK command with, say, a couple of space characters left, you will save the few characters you have plus all the erased text in memory, but you will wipe out the old version of the Text on the disk, which is the Text you wanted to get back.

**How to force load a different disk:** Delete the entire Text, change the disk in the drive, then use the DISK command. SwyftWare will load the new disk.

Another way to discard the Text on-screen and load whatever is in the drive is to hold down the CONTROL (USE FRONT) and Open-Apple keys, and, while holding both of these keys, press and release the RESET button at the upper right keyboard (see Part 22, "Overriding

the *DISK Command*" for a complete discussion of this technique).

### ***Something You Don't Want to Do When Using the DISK Command***

When you use the DISK command, the red in-use light on your disk drive will come on. This is an indicator that the drive is running. *Don't open the disk drive door while this light is on.* It could damage your text. If the disk drive door is left open and you use the DISK command, no harm done; *but don't close the door while the red in-use light is on.* This, too, could damage your text.

Turning off the computer (or having the power fail) during a disk operation can permanently clobber the information on a disk.

### ***Moving a Chunk From Disk to Disk — Copy Up***

If you highlight a chunk of text, change SwyftWare disks, and then use the DISK command, the highlighted text will be automatically inserted at the location of the cursor in the new Text. We say that the chunk of text has been *copied up* from one Text to the other.

For example, say you have part of a report on one disk, and another part on a second disk. You can combine them onto one disk very quickly. Here's how:

1. Do a save (use the DISK command).
2. Highlight the text you want to move.
3. Remove the saved disk from the drive and insert the disk which is to receive the highlighted material.
4. Use the DISK command.

When the second disk loads, a copy of the highlighted text from the first disk will be inserted (and left highlighted) at the location of the cursor. Collapse the highlight, then use the DISK command again to save the combined texts.

Any amount of text may be copied up, from two characters to an entire Text. The only limitation is the space available in the receiving disk. If the chunk won't fit, you get a bleep and nothing happens. You will have to copy up less text or reduce the size of the receiving disk by throwing

something away.

The following illustrations show copying up on the screen.

```
===== 0 =====  
This is paragraph one of our report. We have saved the report on Disk A by  
using the DISK command.
```

```
This is paragraph two of our report. It is a model of brevity.
```

This is paragraph three of our very intriguing report. While you were reading  
this paragraph we switched disks. Now Disk B is in the drive.

```
===== 1 =====
```

*This text is from Disk A. We want to copy up paragraph two, so we highlight  
it after using the DISK command. Next we take Disk A out of the drive,  
insert Disk B, and use the DISK command again. When Disk B loads, a  
highlighted copy of the second paragraph will appear in the text, as shown  
below.*

```
===== 0 =====  
This is a paragraph from Disk B. The cursor was positioned just below this  
paragraph the last time we saved this text.
```

```
This is paragraph two of our report. It is a model of brevity.
```

This, too, is a paragraph from Disk B. As you can see, the highlighted  
paragraph from Disk A was copied up when we loaded Disk B.

```
===== 1 =====
```

*This text is from Disk B, except for the second paragraph, which was  
copied up from Disk A.*

If you accidentally copy up a chunk of text, don't panic. It is very  
easy to get rid of it. Since it is highlighted, just press DELETE. Same  
thing goes if you don't like the position of the copied text: Delete it,  
move the cursor and insert it. When you're satisfied, collapse the  
highlight and save the combined text.

If you accidentally highlight some text, use the DISK command, then  
use the DISK command again, forgetting to switch disks in the drive,  
this merely saves the text a second time without copying up.

## ***Using Belt and Suspenders, or Making a Backup Disk***

Many people feel more secure having a *backup* disk — a second, identical copy of their work — as an insurance policy against possible loss or destruction. If one disk gets munched, then there's always the other.

You can make a second copy at any time onto a blank (fresh out of the box) disk. Put the blank in the drive immediately after you use the DISK command to save onto the first disk. Use the DISK command again. This will create your backup copy.

You can use your backup as an original. As far as SwyftWare is concerned, the original and the copy are identical. You can use the DISK command any number of times on one disk, then put the second disk in the drive, use the DISK command, and SwyftWare will save the Text on the second disk.

*The most important thing to remember is not to do any typing or other operation between the two uses of the DISK command. In other words, the DISK command has to be the last thing you did to the first disk before you save on your backup.*

You can lose the backup relationship between disks if you save on one of them, then turn the computer off before saving on the other. If you now turn on the computer, load the first disk, and try to save on the second disk, you will get a bleep. The same thing can happen if you save on one of the disks, then load another Text without saving on the backup disk.

## ***How to Make a Used Disk Into a Blank Disk***

This is a deadly command, since it destroys whatever is on the disk. After you apply this command, the disk is "new" as far as SwyftWare can tell. First type the dreaded incantation:

CALL 3600

Then highlight it and use the CALC command (USE FRONT G). The disk drive whirs, the deed is done. You can now use the DISK command to save onto this disk. SwyftWare will treat it like a blank. Once executed, CALL 3600 is irrevocable. Fortunately, it is not likely to happen by accident.

## ***Making a Backup Disk After CALL 3600***

Used disks can be wiped clean and then used for backups as if they were brand new. Here's how.

1. Have the Text you want to back-up on the display (let's call this Text the *original* ).
2. Put the used disk (the one you want to wipe clean) in the drive. Type and highlight CALL 3600 and use the CALC command. This disk is now a blank.
3. Take the blank disk out of the drive, press DELETE to get rid of the "CALL 3600" that was left highlighted on the screen.
4. Put the *original* disk in the drive and use the DISK command.
5. Put the blank disk in the drive and use the DISK command.

This method satisfies the rule about not typing or doing any other operation between uses of the DISK command when making backup disks.

## ***Running Apple Software, or, So You Want to Play a Game***

SwyftWare will not stand in the way of any standard Apple program disk you may want to run. Just put it in the drive and use the DISK command. As long as you've saved any changes in the SwyftWare disk you've been using, the Apple program will operate as though SwyftWare wasn't there.

To get back to using SwyftWare after using some other program, remove the Apple program disk, place a SwyftWare-generated disk in the drive, then press and hold the CONTROL and Open-Apple keys, and, while holding them, press and release the RESET button. If you do not have a SwyftWare-generated disk, you may have to turn the computer off and then on again to get back into SwyftWare.

***Note to //e users:*** *If you have a Sider\* hard disk and you want to get back to using SwyftWare, you must have a BASIC prompt*

]  
or  
>

on your display. Put a SwyftWare disk in Slot 6 Drive 1, type PR#6 and tap the RETURN key.

### ***When You Run Out of Room***

You can tell when this has happened since it's the only time that SwyftWare will bleep *every time you tap a character key*. The cursor will not move forward when you type, though it will leap and it will respond if you delete character by character.

This always seems to happen at the most inconvenient moment. But it's easy to fix. You're usually in one of two situations when you run out of room: either you don't mind throwing out some of the text, or you want to save everything you've got and add more to it.

Let's consider the first case first (since we're so methodical).

### ***When You Run Out of Room and Can Throw Something Away***

If you can throw away a few chunks of text, there's no problem. Highlight and delete them and you have some more room. If only one chunk can be thrown away and you highlight and delete it, this *will not get you back the room this chunk took up!* That's because it is still stored, waiting to be INSERTed again.

In two steps, here's how you avoid this problem and get back the room you need:

1. Highlight and delete the text you want to throw out.
2. Highlight and delete two characters. (Leap from character to character, press both LEAP keys, then press DELETE; character-by-character deletion will not do.)

Now INSERT only has two characters to remember, as it has thrown away the big chunk you deleted first. You now have more room. You can retype the two characters you just highlighted and deleted.

## ***When You Run Out of Room and Want to Preserve Everything***

The trick here is to keep part of your text on the *original* disk, and move the rest of the text onto the *new* disk. The way this is done is to make *two* disks with the entire Text, and then erase the *first* part of the text from one of them, and the *second* part of the text from the other. Here's the recipe:

1. Save the text on the *original* (full) disk by using the DISK command.
2. Remove the *original* disk from the drive and put in the blank *new* disk.
3. Highlight and delete the first part of the text.
4. Use the DISK command to save the remainder of the text on the *new* disk.
5. Remove the *new* disk and set it aside.
6. Highlight and delete the entire Text, so that nothing is left (not even one character).
7. Put the *original* disk in the drive.
8. Use the DISK command to bring in the complete text.
9. Highlight and delete the second part of the text (the part you just copied onto the *new* disk).
10. Use the DISK command to save the remainder (first part) of the text on the *original* disk.

This is not the fastest way to do this, but it's the safest, since you have a copy of the entire text around at all times. Once the ten steps are done, you may have to use the technique described in "*When You Run Out of Room and Can Throw Something Away*" to clear INSERT of the big chunk of text.

A faster, though not as safe technique the authors have used is this: Make a *copy* disk by putting a blank disk in the drive and using the DISK command. Now delete the first half of the text and get rid of the stored, deleted text by highlighting and deleting a couple of spaces.

Continue working. Later, when you get around to using the *copy* disk, you have to remember to delete the second part of the text. It is the problem of having to remember what to delete and the chance that you can lose track of what you are doing that makes us recommend the ten-step method given above.

You can also split a text by using copy up, so long as the disk you want to use for the *new* disk already has some text on it. Make a disk with just one character on it if you like, and copy up to that.

### ***How to Split a Text Onto Two Disks When the Text Is Not Full***

Use steps one through ten above. The method works fine even if the text isn't full.

### ***A Strange Case: The Text Is Empty, Yet You Get a Bleep on Every Character You Type***

A rather interesting and paradoxical situation can arise if you've filled up the Text and never deleted a chunk (this might happen during a SEND), or if you have done an INSERT, gotten a bleep and continued typing until the Text is full.

Suppose you now highlight the entire Text and delete it. You will see just two page characters on the screen (0 and 1). It looks like an empty text. But, because the text is really full of deleted characters, every character you type still causes a bleep.

How to get rid of this jammed text? You can't use the usual method because you cannot type two characters to delete. The solution: Use the INSERT command to bring back the deleted text. Use character-by-character delete to remove a few characters. Now delete the whole remaining text. You have room to delete a chunk of two characters now. Type them, highlight them, and delete them.

\* *Sider* is a registered trademark of First Class Peripherals Inc.

## Part 14: The CALC Command

CALC is used when you need to do arithmetic.

### How to Do Calculations

To do a calculation, such as adding 34 and 78, think of asking SwyftWare:

How much is 34 plus 78?

Abbreviate "How much is?" to a question mark and type the formula (followed by a space or return so that the answer, when it appears, will be easy to read). Highlight the formula (including the question mark) and use the CALC command (USE FRONT G). The answer will appear in your text just after the problem:

? 34 + 78; 112

The problem will remain highlighted so that you can delete it, leaving only the answer in your text. A return character is put in after the answer. If you do not want a return after the answer, type a semicolon after the arithmetic expression. For example, ? 34 + 78; will not be followed by a return.

```
===== 0 =====
You can do a calculation anywhere you like. All you have to do is type it,
highlight it, and use the CALC command -- right in the ? 34 + 78; 112 middle
of a sentence, if you wish.
===== 1 =====
```

*Using the CALC command in the midst of text*

### CALC Is an Autoextending Command

When using CALC, you do not have to press both LEAP keys at once in order to highlight the text.

If you use the CALC command after a leap, the text you covered in the leap will highlight automatically and the command will be performed.

If you type something after a leap, after a creep, or immediately after extending the highlight, and then use the CALC command, the highlight will automatically extend to what you have typed and the command will be performed.

### ***Calculations You Can Do***

<b><i>Operation</i></b>	<b><i>Symbol</i></b>
add	+
multiply	*
subtract	-
divide	/
exponentiate	^

Multiplication is symbolized by an asterisk to avoid confusion with the letter "x". You may also use parentheses in expressions.

### ***The Rest of the Details of Arithmetic...***

are in the AppleSoft BASIC manual. When you use the CALC command you are actually writing a BASIC "PRINT" statement. You don't need to know this to perform calculations on SwyftWare, but if you are interested in pursuing the subject, the BASIC manual will show you how to use functions such as LOG, SIN, COS, and TAN. You can write whole BASIC programs in the SwyftWare editor, LIST and RUN them. Even graphics is available — but how to write programs is beyond the scope of this manual.

## **Part 15: CALC Command Miscellany**

In addition to its major use — doing calculations — CALC can perform a few odd jobs, such as telling you how much more you can type before you run out of room, or storing brief stock phrases.

### ***How Much Room Is Left?***

To get the answer, type and highlight

```
?RO%
```

and then use the CALC command. ("RO" stands for "ROom," and the percent character is required by BASIC.)

This will tell you exactly how many characters you can type before the Text is full. If the value comes out negative, this means you have more than 32,768 characters left. A negative answer usually means you have so much room that you don't really care about the exact figure, but if you must know, just type and highlight

```
?65536 + RO%
```

and use the CALC command.

### ***Stored Messages***

CALC can store phrases that you use often but don't like to retype time and again. For example, suppose that you had to include a legal phrase many places in a document. Type

```
L$ = "unless the party of the first part gives prior written notice."
```

just once, highlight it, and use the CALC command (the highlight will not collapse, but the message will be recorded). Now, whenever you type, highlight, and use the CALC command on the expression

```
?L$;
```

the phrase you set up will automatically be inserted into the text (without quotation marks); the semicolon eliminates a return that

would otherwise appear at the end of the stored message. In the example, we used "L" for "legal."

You can have up to 26 phrases stored as letters of the alphabet (A\$, B\$, C\$, ... Z\$), as long as the total does not exceed the space allocated for BASIC. Each phrase can be up to 80 characters in length, must be typed between quotes, and must not contain any returns.

Those of you conversant with Apple's BASIC language will be able to find many other applications, including ways of storing more and longer phrases.

### ***Automatic Save Feature***

You can set a timer within SwyftWare that will save your text automatically if you make some changes and then leave your Apple alone for a while. The timer is set in intervals of roughly 20 seconds. For example, to set your Apple to automatically save the text 20 seconds after your last keystroke, type

```
POKE 5523,1
```

highlight this expression and use the CALC command.

The final number in the expression "POKE 5523,1" is the timer control. The "1" signifies one 20-second interval. A "2" would equal about 40 seconds, and so on. The maximum is 120 (about 2400 seconds, or 40 minutes).

When you start SwyftWare up, the timer control is set to 0. In other words, the timer is off.

## **Part 16: BASIC in the Swyft Environment**

Mixing programming and text editing can be very powerful. For example, to make a calendar for the month of January, you might write the following program

```
10 FOR I = 1 TO 31
20 PRINT "JANUARY "; I
30 NEXT I
RUN
```

If you then highlight it (including the RUN) and use the CALC command you will get a calendar that begins

```
JANUARY 1
JANUARY 2
JANUARY 3
JANUARY 4
JANUARY 5
JANUARY 6
```

and so on. You can even LIST the program by typing the word "LIST" and using the CALC command on it. Almost all BASIC commands work as usual. However, the size of the BASIC program is limited to about 900 bytes (in its compacted internal form). So SwyftWare is not suitable for developing large programs.

Use of BASIC can be dangerous to your text. If used incorrectly or with certain values, CALL, POKE, and PEEK can cause you to lose your text. Graphics — both regular and high resolution — will work in some Apples, but they, too, may clobber your text. Graphic images are stored in the same place as text. High-resolution images will usually not clobber anything if you have only a few thousand characters in your text (a full page or two).

You can try anything safely by doing a DISK command and taking out your disk before you run a BASIC program that might affect your text. Check out your text after running the program, and if everything is OK, then use the DISK command to save your work.

While doing simple arithmetic is generally quite safe, running BASIC programs may wreak havoc with your text. We therefore recommend that you not keep BASIC programs on disks that also have important text on them. If running BASIC does cause the system to "hang," try CONTROL-RESET. This will often get things up and running, and usually doesn't cost you the loss of your text (see note below). Backup disks are a *great* idea when experimenting with the limits of BASIC.

Any programs that you write are run in the same manner you do calculations: leap from one end of the program to the other, then use the CALC command. The highlight autoextends.

**Note:** *While a BASIC program is running, the traditional CONTROL C (or USE FRONT C) can be used to interrupt it. You may have to use it a few times to stop a program. The computer may insert the word "BREAK" into your text when you do this. A neater way to stop the program is to press and hold the CONTROL key, and, while you hold it, press and release the RESET key (located in the upper right corner of the keyboard).*

To get into Applesoft BASIC as though SwyftWare were not there, do an Apple-style boot (CONTROL-OPEN-APPLE-RESET) with an empty drive, and as soon as the disk light comes on, use CONTROL-RESET. To get back to SwyftWare you will have to reboot with a SwyftWare disk or, in the case of SwyftCard, turn the power off and on.

**WARNING:** *Don't allow a BASIC statement to cause the text to consist of more than 200 pages. This will cause you to lose your text.*

**For experts only:** To exit SwyftWare and enter BASIC, put a BASIC disk in the drive and use the DISK command. If you don't have a BASIC disk and you still want to enter BASIC, make sure there's no disk in the drive, press CONTROL-OPEN-APPLE-RESET and wait for the disk drive light to come on, then press CONTROL-RESET, and you'll be in BASIC. However, you will not have DOS or ProDOS since you didn't boot.

## ***Chapter IV***

### ***Printing and Communicating***

<i>Part 17</i>	<i>Printing</i>	<i>57</i>
<i>Part 18</i>	<i>Formatting the Printout</i>	<i>60</i>
<i>Part 19</i>	<i>Detailed Control of Your Printer</i>	<i>66</i>
<i>Part 20</i>	<i>The SEND Command (Electronic Communications)</i>	<i>71</i>



## **Part 17: Printing**

### **Printer Preliminaries**

It is very easy to print with SwyftWare. However, in the name of progress, all printer manufacturers like to do things their own way, and as a by-product of all this progress nearly every printer requires a custom setup. So while it is very easy to print with your SwyftWare, it may or may not be easy to set up the printer and the printer interface.

In *Part 19* we list the setups of over 60 popular printers. If your printer isn't on the list, you may have to figure out the settings from the manuals for your interface card and printer. *Part 19* explains how the settings were constructed. If you purchased your equipment from a dealer, it should be in the dealer's interest to make sure that it works.

*Note: If you are using an Apple //e, be sure that the printer interface card is in Slot 1.*

### **How to Print**

You can print whatever is highlighted by using the PRINT command (USE FRONT N). Actually, you can skip the highlighting: just leap from first character to last (or last to first) of what you want to print, then use the PRINT command. That's it. The highlight will extend automatically and printing will start. Jef (see cover) always highlights anyway to make sure he's printing exactly what he wants.

You can print as little as one character, a part of a word, a few words, or as much as you want: a paragraph, a letter, a chapter of a novel, or the entire Text. You can start printing in the middle of a line, or even in the middle of a word, and you can choose the ending point with equal flexibility.

If you begin the highlight in the middle of a line, then your printer will start to print in the middle of a line. The way it looks on the screen gives a very good indication of how it will look on paper. The highlight even vanishes from the screen line-by-line at the same rate the text goes to the printer (or at the rate it goes to a print buffer, if you have one).

The page breaks and page characters in your text will correspond to the perforations between sheets of paper in your printer (top and

bottom margins will be inserted). You may have to experiment to find just the right spot to position your paper so that the top and bottom margins come out equal.

If your printer requires that you print one page at a time, just highlight one page at a time and print it. David (see cover) finds it easier to go through the text and type a page character one line above each page break (the page break will disappear when you type a page character above it). After each page prints out, leap forward to the next page character and use the PRINT command.

### ***Printers and Form Feeds***

A form feed is a special signal you can send to some printers that tells them to move to the next page. Page breaks and page characters send a form feed to the printer, causing it to roll out the rest of the paper page it's working on and get ready to start a new page.

However, if the first character in the chunk of text you've highlighted is a page character, then that particular form feed is not sent to the printer. This saves a lot of paper, as it means that you won't lose a sheet of paper if you highlight from one page character to another when you are about to print (see *"How to Print a Document"* below).

If your printer requires you to eject a blank sheet of paper before you start printing, just put two page characters at the beginning of what you want printed and include both of them in your extended highlight.

### ***How to Print a Document***

In the tutorial we recommended that you begin a document by typing a page character, then let the page breaks appear automatically as you typed the text. This means that many documents will have only two page characters: one at the beginning and one at the end. To print the document, leap from one to the other, then use the PRINT command.

If the document has several page characters in it, use the LEAP AGAIN key to help the cursor leap from first to last, then use the PRINT command.

### ***How to Print the Entire Text***

There's a trick to this: Leap to the first page character in the Text, the one with a number zero in it. Let go of all keys. Press and hold LEAP BACKWARD, then tap the PAGE key. Let go of all keys and use the PRINT command.

When you tap the PAGE key you will "hop the back fence" and leap to the last page character in the text. You've leaped from the very beginning to the very end, so when you use the PRINT command, SwyftWare will highlight and print the entire Text.

### ***Getting the Page Numbers Right***

SwyftWare automatically leaves page one unnumbered, then prints a number for page two and higher at the bottom center of each page. In *Part 18* we explain how to play with pagination in detail. But without ever reading that section you can easily get any document to begin with page one: just move the document to the front of the text.

Here's how: Delete the document, leap the cursor to the page character with the number zero in it, and use the INSERT command. The inserted text (your document) will be highlighted, so you can use the PRINT command as soon as it appears on-screen (which is pretty soon indeed).

### ***Emergency Stop Print***

To halt a print that is in progress, press and hold the USE FRONT key, and, while you hold it down, press and release the RESET button. Do not press either the Open-Apple or Solid-Apple keys while you do this. The unprinted text will remain highlighted. To resume printing, use the PRINT command. Some printers may require you to move the cursor back to the nearest page break before you begin to print.

## ***Part 18: Formatting the Printout***

SwyftWare automatically formats your text and printout with standard margins, single-spaced lines, and 66-line pages. Changes can be made in the format by typing a brief formula and using the CALC command. The change is recorded on the disk with the rest of the Text when you do a save (use the DISK command). Remember that the autoextending highlight feature is available with the CALC command (see *Part 10, "Autoextending Highlight"*).

### ***Line Width***

Both the screen and printer are normally set up for 80 characters per line. The Apple display is unable to show more than 80 characters per line, so it is impossible to set line width at greater than 80 characters and retain "what you see is what you get" printing. To set the line width to a lower number, say 66 characters, type

```
WL%=66
```

Highlight this formula, then use the CALC command. After a second or two, the lines on-screen will narrow to 66 characters and the entire Text will be automatically reformatted, including pagination. The highlight remains extended so that you can tap the DELETE key and erase the formula from your text.

You can set the line width to any value from 16 to 80. Values too large or too small are set to the maximum or minimum values respectively. "WL" stands for "Width."

Narrowing the line width increases the number of pages in the Text. More than 200 page breaks might destroy your text, so before SwyftWare changes the width, it approximates roughly how many pages will be added, tending to exaggerate the number. If it appears that the total number of page breaks will exceed 200, SwyftWare won't change the margins. However, with determined work you may be able to fool it.

### ***Left Margin (of the Printer)***

The left margin is normally set at the left edge of the printer. It can be moved to the right a given number of characters, say 16, by typing,

highlighting, and using the CALC command on

`MA%=16`

This does not affect the computer's display in any way. "MA" stands for "MArgin."

The whole body of printed text moves to the right on the page when you increase MA%, thus you may wish to reduce WI% when you increase MA%. For example, if you set MA%=12 you may want to reduce WI% from 80 to 68 to keep the same right-hand margin. Narrow, 80-column printers may not be able to print text with the sum of MA% and WI% greater than 80. Some will not operate, and others may break up lines arbitrarily.

### ***Top and Bottom Margins***

The top margin is set by typing, highlighting, and using the CALC command on an expression that gives the number of blank lines you want in the top margin. If you wanted six blank lines above the printout on each page, you would type

`AB%=6`

highlight it, and use the CALC command.

Most printers print six lines per inch, thus AB%=6 provides the standard one-inch top margin. "AB" stands for "ABove."

Similarly, typing

`BE%=6`

highlighting, and using the CALC command sets the bottom margin to six blank lines (BE% is automatically set at 6 when SwyftWare starts up). "BE" stands for "BElow."

Unless you change AB% or BE%, SwyftWare will automatically print six-line (typically one-inch) margins above and below your text.

When you increase the margins, the number of lines per page automatically decreases to compensate; when you decrease the margins, the number of lines per page automatically increases.

SwyftWare assumes that there is room for a total of 66 lines per page from the top edge to the bottom.

### ***Number of Lines Per Page***

SwyftWare normally allows for 66 lines per page (top to bottom). A lesser number, say 34, may be selected by typing

`PL%=34`

highlighting this expression, and then using the CALC command. "PL" stands for "Page Length."

Setting PL% to a large number (255 is the maximum) results in long stretches of unpaginated text. Reducing the page length usually increases the number of pages in the Text. More than 200 page breaks might destroy your text, so before SwyftWare carries out a change in PL%, it makes an exaggerated approximation of how many pages will be added. If SwyftWare guesses that the total number of page breaks will exceed 200, it won't change the page length.

When SwyftWare refuses to set PL% as low as you want, try approaching it in steps. For example, if PL% won't change from 66 to 16, try 20 first, and if you get that, then try 16.

### ***Lowest Page Number***

The first page of a document or letter is customarily unnumbered; numbering typically commences on page two. SwyftWare does this automatically (PF%=2 when you start up). You might desire something different, like having the page numbering start with "1". In that case you would type and highlight

`PF%=1`

and then use the CALC command.

If you wanted page numbering to start on page 3, you'd use PF%=3, and so on. You can use a very high value (any value above the highest page number in your text) to get rid of all printed page numbers. The maximum value for PF% is 32767. Though the numbers will not be printed, they will still appear on your display. "PF%" stands for "Page Floor."

## ***Changing Page Numbers***

Consecutive page numbering is possible for very long documents that sprawl over many disks. For example, if the first disk has page numbers 1 through 22, how do you get the second disk to start with page 23? You simply set the number on the first page character to 22 (the first page character is normally "0"; that is, PA%=0 when you start up). That way the page number at the bottom of the first page will have the desired value of 23. To do this, type and highlight

```
PA%=22
```

and then use the CALC command. "PA" stands for "Page."

When the text on one disk includes many documents, only the first document starts with page one. All other documents will start out with some other page number. One way to alter the page numbering of such documents is to move them to the front of the text, as explained in *Part 17, "Getting the Page Numbers Right."* Another way is to move the zero page break ahead in the text. This is done by resetting the page numbering to a negative number.

For example, suppose you have a letter that begins on page ten. You'd prefer that this letter began on page one, so you type and highlight

```
PA%=-9
```

and use the CALC command. This reduces page character 9 to 0. Page 10 is now page 1 (the page numbering on your display will change, so you can check to see if you've got it right). Some documents may have negative page numbers when you use this technique of changing the page number. This is OK. Since the normal setting of PF% prevents page numbers less than 2 from printing, these negative numbers simply will not print.

To go back to normal numbering, type and highlight

```
PA%=8
```

and then use the CALC command. "PA" stands for "PAGE."

## ***Changing the Vertical Position of the Page Number***

The position of the page number is normally dead center, three lines from the bottom of the page. While the page number will always be centered, it can be moved up or down. The lowest setting is zero, which causes the page number to print on the bottom line of the page. The highest is limited to two less than the number of lines in BE%, the bottom margin. If you wanted to move the page number to the fourth line from the bottom of the page, you would type

LP%=4

highlight it, and use the CALC command. You cannot change the horizontal position of the page number. "LP" stands for "Latitude of Page number." (LP%=3 is normal.)

## ***Double-spacing***

To make your text print double-spaced, type and highlight

SP%=2

and then use the CALC command. The new setting will not noticeably affect the appearance of the text, which remains single-spaced on the screen. The page breaks and page characters will move closer together because you have half the number of lines per page. "SP" stands for "SPacing."

To return to single-spacing, type, highlight, and use the CALC command on

SP%=1

This is the normal value for SP%.

SP% can be 1 or 2. Values less than 1 will be changed to 1; values greater than 2 will be changed to 2.

The text on the display may jump when you use the CALC command to change the line spacing.

Use caution in changing to double-spacing if the text includes more than 100 pages. This may cause you to have more than 200 pages,

which could cause your text to be lost (from the screen, not from the disk).

### ***Questioning the Printer Values***

If you want to know how many characters you have per line, type

```
?WI%
```

highlight it, and use the CALC command. If you want to know what number the first page character is set to, type

```
?PA%
```

highlight it, and use the CALC command. All the values described in this chapter may be questioned in this manner.

### ***Resetting the Format Values to Normal***

The standard default values for each of these settings can be found in the List of Features (default values are normal values). To adjust any setting back to the standard default value, type the expression, an equal sign, and the default value, then highlight it and use the CALC command.

For example, to set the line width back to the normal value of 80, you would type

```
WI%=80
```

and use the CALC command.

### ***Adjusting Several Format Values at Once***

You can adjust several values at once by typing a colon between them. For example, to reset line width, page numbering, and top margin, you would type

```
WI%=80:PA%=0:AB%=5
```

and use the CALC command.

## **Part 19: Detailed Control of Your Printer**

Though most printers require some setup, SwyftWare often works reasonably well without it. Test the result of using the PRINT command first. If you get the wrong font or inappropriate margins, or if all you get is garbled characters, lots of printing all on the same line, double-spacing (an extra space between lines of print when you didn't ask for it), or nothing at all, then read on.

We cannot cover the installation of your printer, printer interface and printer cable, nor explain how to load the paper. We leave that up to the printer and interface card manuals. Please read them carefully or have your dealer or an experienced friend lend a hand.

### **Apple //e Interface Cards**

We tested parallel printers with the Grappler+\*, the Apple parallel card, and the Slot-Buster Card\*, and tested serial printers with the Apple Super Serial Card. The switches on the Grappler+ card were set as follows:

1	2	3	4
<i>off</i>	<i>on</i>	<i>on</i>	<i>on</i>

The Slot-Buster Card has interfaces for parallel and serial printers and also has speech capability. For use with parallel printers we set the switches as follows (*dn* is down or off, *up* is up or on):

1	2	3	4	5	6	7	8
<i>up</i>	<i>up</i>	<i>dn</i>	<i>dn</i>	<i>dn</i>	<i>dn</i>	<i>dn</i>	<i>dn</i>

Super Serial Card switches were set as follows:

<b>SW1</b>	1	2	3	4	5	6	7
	<i>dn</i>	<i>dn</i>	<i>dn</i>	<i>up</i>	<i>dn</i>	<i>up</i>	<i>up</i>

<b>SW2</b>	1	2	3	4	5	6	7
	<i>dn</i>	<i>dn</i>	<i>dn</i>	<i>up</i>	<i>dn</i>	<i>dn</i>	<i>dn</i>

\**Grappler+* is a registered trademark of Orange Micro Inc.  
\**Slot-Buster* is a registered trademark of RC Systems Inc.

If there is a switch on the parallel card which supplies a "line feed" for each "carriage return" (sometimes called *auto LF with CR*) it should be on. If this gives you double-spacing (extra space between lines of print), then it should be off. The Grappler+ doesn't have such a switch, therefore you will need to adjust your printer initialization message by setting the first code to a 1 (see "*Printer Initialization*" section below). You will need to make this adjustment even if you have a printer listed in *Group 1, "Printers Not Requiring Any Set Up,"* below.

### ***Printer Switches***

Printers often have a multitude of switches, and in places that sometimes require disassembly to access. We have designed SwyftWare to work with most printers *with the switch settings as they come from the factory*. So don't alter their settings unless you find some reason that makes it necessary. If you have a serial printer with a baud rate setting, set it for 9600 baud.

### ***Printer Initialization***

If the printer does not operate satisfactorily when you use the PRINT command, then follow the instructions here and in the following sections.

To initialize your printer (get it ready to go) you must send it a printer initialization message. This message tells the printer how you want it to work. On the following pages we provide the printer initialization messages for more than 60 different printers. You can use these just as they are. We will also explain how these settings are created. This will be of interest if you just want to know more, if you want to make adjustments, use special features such as italics or bold, or if you want to use a printer that's not on the list.

Since new printers come on the market all the time and past generations of printers remain in use, we may not have listed your printer brand or model. It would not hurt to try some of the initialization messages listed in this chapter, especially if one of them is for your brand but a different model.

### ***How to Use the Printer Initialization Message***

Once you have the message for your printer, all you need to do is type it, highlight it, and use the CALC command. Then, whenever you use the PRINT command, the printer initialization message will be sent to the printer ahead of the text you want printed.

You can type the printer initialization message *wherever you happen to be in the text* — in the middle of a paragraph if you like. Just highlight it and use the CALC command.

**Note:** *If your printer keeps on printing over and over on the same line but otherwise seems OK, then change the part of the messages below that say*

```
PR$=CHR$(0)+
```

*into*

```
PR$=CHR$(1)+
```

*and leave the rest the same as shown.*

### ***When Line Breaks Appear Where They Shouldn't While Using the Apple IIc***

You may find lines breaking at odd places in your printout when using an Apple IIc computer to run your printer. This problem can be corrected with a small addition to your printer initialization message.

**Example:** *Suppose you have an Apple ImageWriter and erroneous line breaks are cropping up. You retype the printer initialization message (found in the "Group 2" listing beginning on the next page) adding a new line as shown below:*

```
PR$=CHR$(0)+CHR$(9)+CHR$(5)+CHR$(27)+CHR$(69)
PR$=PR$+CHR$(9)+"132N" ← new line
US$=CHR$(27)+CHR$(88)
UE$=CHR$(1)+CHR$(27)+CHR$(89)
```

*You highlight the four lines and use the CALC command. This gives your Text a new printer initialization and your printer should run fine now.*

If you find that lines break when using your IIc, then whatever the printer initialization message is for your printer, retype it, adding the expression

```
PR$=PR$+CHR$(9)+"132N"
```

in the second line as shown above. Highlight the new expression and use the CALC command.

## ***Printer Settings***

### ***Group 1: Printers Not Requiring Any Set Up***

Brother M-1009  
Brother Twinriter 5 (DP dot matrix mode)  
Epson FX-80, FX-80+, FX-85, FX-100, RX-100, FX-185  
Hewlett-Packard ThinkJet  
IBM Graphics Printer  
Panasonic P-1092  
Silver Reed EXP-500  
Smith-Corona D-100, D-200, D-300  
Spectrum JX-80, LX-80  
Star Micronics SB-10, SD-10, SD-15, SG-10, SG-15, SR-10, SR-15

If you want to change back to using one of the printers above after working with some other printer, type, highlight, and use the CALC command on

```
PR$=CHR$(0)+CHR$(0)+CHR$(5)+CHR$(27)+CHR$(77)+CHR$(27)+CHR$(100)+CHR$(0)
US$=CHR$(27)+CHR$(45)+CHR$(49)
UE$=CHR$(1)+CHR$(27)+CHR$(45)+CHR$(48)
```

### ***Group 2: Printers Requiring Set Up (alphabetical by manufacturer)***

Apple Imagewriter, Apple Scribe

```
PR$=CHR$(1)+CHR$(9)+CHR$(5)+CHR$(27)+CHR$(69)
US$=CHR$(27)+CHR$(88)
UE$=CHR$(1)+CHR$(27)+CHR$(89)
```

Apple Laserwriter

```
PR$=CHR$(1)+CHR$(0)+CHR$(0)
US$=CHR$(27)+CHR$(69)
UE$=CHR$(1)+CHR$(27)+CHR$(82)
```

Brother 2024

```
PR$=CHR$(0)+CHR$(10)+CHR$(5)
US$=CHR$(27)+CHR$(45)+CHR$(49)
UE$=CHR$(1)+CHR$(27)+CHR$(45)+CHR$(48)
```

Brother HR-10, HR-15XL, HR-25, HR-35,  
Twinriter 5 (WP daisy wheel mode)

PR\$=CHR\$(0)+CHR\$(4)+CHR\$(5)  
US\$=CHR\$(27)+CHR\$(69)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(82)

C-Itoh (Apple parallel printer)

PR\$=CHR\$(0)+CHR\$(9)+CHR\$(5)+CHR\$(27)+CHR\$(69)  
US\$=CHR\$(27)+CHR\$(88)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(89)

Data Products 8010, 8050

PR\$=CHR\$(0)+CHR\$(8)+CHR\$(0)+CHR\$(30)  
US\$=CHR\$(27)+CHR\$(102)+CHR\$(49)+CHR\$(36)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(102)+CHR\$(49)+CHR\$(36)

Diablo 630

PR\$=CHR\$(0)+CHR\$(4)+CHR\$(5)  
US\$=CHR\$(27)+CHR\$(69)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(82)

DTC 380 (and most other DTC models)

PR\$=CHR\$(0)+CHR\$(4)+CHR\$(5)  
US\$=CHR\$(27)+CHR\$(69)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(82)

Epson LQ-1500, SQ-1500

PR\$=CHR\$(0)+CHR\$(9)+CHR\$(5)+CHR\$(27)+CHR\$(33)  
US\$=CHR\$(27)+CHR\$(45)+CHR\$(49)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(45)+CHR\$(48)

Fujitsu Dot Matrix 24

PR\$=CHR\$(0)+CHR\$(0)+CHR\$(5)+CHR\$(27)+CHR\$(77)+CHR\$(27)+CHR\$(100)+CHR\$(8)  
US\$=CHR\$(27)+CHR\$(69)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(82)

## Hewlett-Packard LaserJet

PR\$=CHR\$(1)+CHR\$(8)+CHR\$(8)  
US\$=CHR\$(27)+CHR\$(36)+CHR\$(100)+CHR\$(68)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(38)+CHR\$(100)+CHR\$(64)

## Integral Data Systems 480

PR\$=CHR\$(8)+CHR\$(8)+CHR\$(5)+CHR\$(30)+CHR\$(27)+CHR\$(82)+CHR\$(44)+CHR\$(49)  
PR\$=PR\$+CHR\$(44)+CHR\$(36)  
US\$=CHR\$(95)+CHR\$(8)  
UE\$=CHR\$(8)

## Juki 6100, 6300

PR\$=CHR\$(8)+CHR\$(6)+CHR\$(5)  
US\$=CHR\$(27)+CHR\$(69)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(82)

## Mannesman Talley 160L

PR\$=CHR\$(8)+CHR\$(9)+CHR\$(5)+CHR\$(27)+CHR\$(91)+CHR\$(53)+CHR\$(121)  
US\$=CHR\$(27)+CHR\$(91)+CHR\$(52)+CHR\$(109)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(91)+CHR\$(48)+CHR\$(109)

## NEC Spinwriter (all models)

PR\$=CHR\$(8)+CHR\$(9)+CHR\$(5)+CHR\$(27)+CHR\$(35)+CHR\$(65)+CHR\$(27)+CHR\$(73)  
US\$=CHR\$(27)+CHR\$(45)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(39)

## Octave Systems (DTC) Laser Printer

PR\$=CHR\$(8)+CHR\$(4)+CHR\$(5)  
US\$=CHR\$(27)+CHR\$(69)  
UE\$=CHR\$(1)+CHR\$(27)+CHR\$(82)

third line (beginning UE\$=).

### ***Control of the Printer in General***

Ideally, SwyftWare should print perfectly without adjustments. If adjustments have to be made, it would be easiest to make them by throwing a switch on the interface card or the printer. But sometimes that isn't possible without interfering with your other programs. In that case, you need to control the printer through the printer initialization message.

Each item in the printer initialization message is just a number (techies will call them ASCII codes). Because of the way BASIC is designed, each code has to be put between the parentheses in the formula `CHR$( )`. Then all the codes are strung together one after the other by means of plus signs (as was done in the listings above). The resultant message packet for the printer is called `PR$` (PR stands for "PRinter;" and the dollar sign is required by BASIC).

The first code tells SwyftWare whether your printer needs a line feed for each carriage return in order to separate lines. "0" means none is needed, "1" means one is needed. If the printer prints over and over on the same line or if the printing is double-spaced, first make sure that the interface card auto-line feed switch is on (if you have a //e) and the printer's auto-line feed switch is off, then play with the first code in `PR$` until it works right. There are only two choices (0 and 1), so it won't take long.

The second code controls the number of spaces the printer will skip before it prints the first character on each line. Thus this code can be used to set the left margin. In addition to the second code, you can use the `MA%` command and the left margin command of the particular printer you are using to control the left margin (left margin command is discussed in the next section). It's probably easiest to experiment by setting the print string to

```
PR$=CHR$(0)+CHR$(0)+CHR$(0)
```

or

```
PR$=CHR$(1)+CHR$(0)+CHR$(0)
```

and adjusting the second code until the left margin looks right.

The third code is used to align the perforations in your printer paper with the tear-off bar. Whenever a page break or page character is

printed, the printer rolls off enough paper to get to the start of the next page (this is called a *form feed*), plus the number of blank lines specified by the third code.

The third code allows you to compensate for the different distances between the print head and the tear-off bar on various printers. The third code and the top margin are interrelated. The top margin is the space between the perforations and the first printed line. This, of course, is not always the same as the distance between the print head and the tear off bar. Thus you need the AB% command to set the top margin. After the printer completes the form feed plus the line feeds for the third printer code, it adds enough blank lines to complete the top margin. *The third code must always be less than or equal to AB%.* If it is greater, there will be too many lines in the top margin. Both the top margin and the third printer code are best decided by experiment and your personal preference.

**Example:** *Suppose your printer prints over and over on the same line, while the print itself needs to be moved 7 spaces to the right. The tear-off bar is 5 lines from the print head. Thus your PR\$ would begin*

```
PR$=CHR$(1)+CHR$(7)+CHR$(5)
```

*This is often enough. These three values are used by SwyftWare to generate the appropriate returns and spaces, which are sent to the printer. The three values themselves are not sent to the printer.*

But some printers require more, so please read on...

### **Control of Printing Style**

Some printers will give you different print qualities (print size, draft or correspondence quality) or styles (italics, Gothic, bold). You set the style by sending number of characters to the printer each time you use the PRINT command. These characters form the rest of the PR\$ message that we started above.

You have to first determine what characters the printer requires (from the printer manual), and then translate what it says into ASCII codes. For example, if the manual says that you have to send the printer ESC, I, T to get italics, then you would translate this into

```
CHR$(27)+CHR$(73)+CHR$(84)
```

**Note to BASIC experts:** You can, of course, use CHR\$(27) + "IT" in this example, but for the sake of uniformity we use the CHR\$ technique throughout.

It is easy to find the ASCII codes for any symbol you can type. You can ask SwyftWare by putting the symbol (letter, number, space, whatever) in between the quotes of

```
?ASC(" ");
```

highlight it, and use the CALC command. The only other one you usually need to know is that ESC translates to 27. If "control" characters are called for, then CONTROL A is 1, CONTROL B is 2, and so on through CONTROL Z, which is 26.

Once you have found the desired print quality code in the manual and have translated it into its ASCII equivalent, then you would add it onto the general printer controls. In our example you would have

```
PR$=CHR$(1)+CHR$(5)+CHR$(7)+CHR$(27)+CHR$(73)+CHR$(84)
```

And you are done with the PR\$ part.

### **Controlling Underlining (or Local Font Selection)**

If you use underline characters in your text, you have to tell your printer how to handle them. When SwyftWare encounters an underline character (   ) during printing by the PRINT command, it sends an "underline start" message to the printer. Unfortunately, different printers require different forms of the message. This message is called US\$, which stands for "Underline Start" (the dollar sign is required by BASIC). The underline character in your text is printed as a space, thus keeping the arrangement of text the same on the printer as it is on the screen.

When a second underline character (or when two returns or a page character) is encountered during printing, a UE\$ or "Underline End" message is sent to the printer. This underline character, if present, is also turned into a space upon being printed.

US\$ is a message that tells the printer that underlining will start with the next character. The message is built up just as for PR\$, using information from the printer manual, translating into ASCII codes, and using CHR\$ as before.

The first code in UE\$ tells SwyftWare whether the printer is of the kind that underlines a group of characters all at once (in which case the code is CHR\$(1) ) or if it is the kind that underlines a character at a time (in which case the code is CHR\$(0) ). The rest of UE\$ is a message that tells the printer to stop underlining. If your printer settings are not included in the list in Part 19, you will need to translate them from your printer manual.

You can use the underline character in text to turn on or off any enhancements that can be coded in five or fewer codes. Not only underlining, but any other print feature can be used: italics, boldface, letter-quality printing, or even graphics that can be operated by start and stop codes. Though it would be quite a tedious process, you could even print out portions of your text (say, a paragraph at a time), changing the US\$ and UE\$ messages in between so that you get underlining here, boldface there, and italics in yet a third place in the same document.

### ***Debugging Your Printer***

If you've tried everything and you still can't get your printer to work, make sure the cable is hooked up right (the //c has identical-looking ports for printer and modem) and the printer is on-line (the *on-line*, *ready*, *enabled*, or *select* light on the printer is lit).

If both these are OK, try turning off the printer and the computer and starting all over. Sometimes a printer will "remember" the codes you sent it earlier (perhaps while you were trying to get it to work, or while you were using some other program), and will not correctly accept the new information until it has been turned off and on again.

If all else fails, read the printer and interface manuals, and see if your printer dealer will help. In rare cases the problem may be a malfunction of the printer, the cable, or the cards in your computer. Only hands-on testing by a repair person will uncover a problem such as this.

### ***How to Start a New Disk for Your Printer Without Having to Redo Printer Initialization***

The following steps allow you to get a new disk ready to go without having to retype the printer initialization message:

1. With the text of an already initialized disk on the screen, remove the disk from the drive and insert a blank disk.

2. Highlight and delete the entire Text. Type a few spaces, then use the DISK command.

The printer initialization will be saved on the blank disk along with the Text (a few blank spaces).

When a disk already has text on it and you need to change the initialization message, you can still get around retyping by copying up from another disk. You might keep printer initialization messages (and any other boilerplate you may need, such as letterhead or legal phrases) on one disk. Write-protect that disk and copy up from it.

## **Part 20: The SEND Command (Electronic Communications)**

The SEND command allows you to communicate electronically, either through cables to nearby connections or all over the world via the telephone network. You can communicate with data services, bulletin boards, friends, or any "on-line" system. This command is also your key to sharing information with other Apples (whether they have SwyftWare or not), and computers both small and large of other brands, so long as they have a standard modem and communications capabilities.

### **Hardware Requirements**

The SEND command assumes that you have either an Apple //c or an Apple //e with a Super Serial Card. If you are using an Apple //e, the Super Serial Card must be in Slot 2. While you can communicate directly between two SwyftWare-equipped Apples (if they are close enough for you to run a cable between them), this section deals primarily with communications over telephone lines, which means that you also need a modem connected to your Super Serial Card (//e) or serial output (//c).

The modem must, of course, be connected to a telephone line. The modem manual will explain how.

### **Setting Up the Super Serial Card on the Apple //e**

There are two banks of switches, labeled SW1 and SW2 on the Apple Super Serial Card. They should be set as follows (*dn* means the switch is down or off, *up* means up or on):

<b>SW1</b>	1	2	3	4	5	6	7
	<i>up</i>	<i>dn</i>	<i>dn</i>	<i>up</i>	<i>up</i>	<i>up</i>	<i>up</i>
<b>SW2</b>	1	2	3	4	5	6	7
	<i>up</i>	<i>dn</i>	<i>dn</i>	<i>up</i>	<i>dn</i>	<i>up</i>	<i>dn</i>

The only other setting is a black block with a triangle on it. The triangle should be pointing up, toward the word "modem." The Super Serial Card manual explains the use of each switch.

## **Setting Up the Modem**

The switches on an Apple 300 or 1200 modem should be set as follows:

1	2	3
dn	up	up

The Hayes Smartmodem switches (behind the front panel) may be set:

1	2	3	4	5	6	7	8
dn	up	dn	dn	up	dn	up	dn

The Hayes manual explains the use of each switch.

The Hayes Smartmodem 2400 does not have switches. To operate this and other products not mentioned, please consult the manual that came with the modem.

## **The SEND Command Itself**

Compared to setting up, the SEND command is simplicity itself. Type and edit the text you want to send, highlight it, and use the SEND command (USE FRONT D). Skip the highlighting step if you like. Just leap from one end of what you want to send to the other and use the SEND command. The highlight will extend automatically.

Autoextend is especially convenient when you are having an electronic conversation (that is, where you and the device you are communicating with work alternately, as in a normal human discussion). Just type what you want to send, then use the SEND command. It is not necessary to highlight, since everything you typed since the last message was received will be highlighted and sent automatically when you use the SEND command.

If you use a LEAP key in the course of typing your response, you will have to leap from end to end of the whole response before you send it. Use of character-by-character delete is OK: it will not affect the autoextending highlight.

You can type even while a message is coming into the text. It will probably be easier to read what you are typing if you type a return first.

**Note:** Avoid leaping while a message is coming in. As soon as you let go of the LEAP key, the incoming text will begin to spout from where the

*cursor has landed. This can be very confusing, as the incoming text will mix with the text already on-screen.*

The above instructions are all you need to know if you are communicating between two Apple computers equipped with SwyftWare. Working with other systems sometimes requires a bit more effort.

### ***Systems That Require a Return (or Something Else) at the End of Every Line***

Many non-SwyftWare systems require a return at the end of every line. You can do this by typing a return at the end of every line before you SEND. This becomes tiresome, however, and you lose the advantage of automatic word wrap. There's an easier way: if you type, highlight, and use the CALC command on

```
LE%=29
```

then a return will be sent at the end of every line, and after the last character sent as well. This is what most systems expect.

If your text has a return, SwyftWare will use that return and not add one of its own.

To turn off the LE% feature, type, highlight, and use the CALC command on

```
LE%=-1
```

### ***Expert's Use of LE%*** ***(this section can be ignored by almost everybody almost all of the time)***

LE% can be used to send any one or two characters at the end of every line. To send one character, set LE% to the ASCII code for that character. To send two characters (e.g. CR and LF, with ASCII codes 13 and 10) you set LE% to the sum of the first character and 256 times the second character. In the case of CR and LF, you would type, highlight, and use the CALC command on

```
LE% = 13 + 256 * 10
```

If the last character in the line of text matches either character of LE%, one of the characters is dropped so that you don't get doubled characters. Internally, SwyftWare uses ASCII 29 for a return, rather than

ASCII 13. In communicating with the outside world, of course, the standard value of 13 is used. Thus, if LE% is set to 13, you will get two returns for each one in the text.

### ***Dialing***

If your modem will dial for you, read your modem manual to see what message you must send to make it do this. For most Hayes-compatible modems it is important to first set LE%=29. Then you can dial a phone number — say, 555-1212 — by typing, highlighting, and using the SEND command on

```
ATDT 555-1212 return
```

A return must be typed after a phone number command such as "ATDT 555-1212" unless you have set LE% to add a return at the end of each line. This is because every command you send to a Hayes or Hayes-compatible modem must end in a return.

By putting a dialing code in front of every phone number, your phone number list and SwyftWare can be turned into an automatic dialing machine! Just leap to the name, highlight the dialing code and the number, and use the SEND command. When you hear the phone on the other end ringing, pick up the handset and talk.

### ***Answering the Phone***

This, again, is a function of your modem and not of SwyftWare. If you use the settings shown above for a Hayes modem, or if you are using an Apple modem, the modem will automatically answer the phone.

### ***Setting Baud Rates***

***(useful information for many people, though not all)***

For each of the following baud rates, type, highlight, and use CALC on the expression on the right. ("SE" stands for "SEnd.")

300	SE%=5641	<i>(default baud rate)</i>
1200	SE%=6153	
2400	SE%=6665	
4800	SE%=7177	
9600	SE%=7689	
19200	SE%=7945	

The baud rate you choose depends on the requirements of the system you are communicating with, and what your modem can accept. You should set the baud rate before you begin communications. When you save your disk, the baud rate and all other settings are saved, so you may never have to do it again.

SwyftWare normally works at 300 baud, with 1 start bit, 1 stop bit, 8 data bits, and no parity. It doesn't matter if you don't understand what any of these things are, just that they match what the service you are using expects. We have found that the choice we have made will work with most services and bulletin boards, and you can ignore the whole business if it works for you the way it comes.

### ***Setting Two Stop Bits and Other Fine Points*** *(a section for experts only — most people can and should skip this)*

The BASIC variable SE% is used to initialize the Super Serial Card for its communication protocol. This operation is analogous to initializing the printer with PR\$. By changing the value of SE% and modifying the modem protocol you can communicate with a wide variety of data services.

Here's how to find the value for almost any settings you might want. First, find the number for the baud rate:

***Table 1: Baud Rates***

<b><i>Rate</i></b>	<b><i>Number</i></b>	<b><i>Comment</i></b>
153600	0	<i>yes, we do mean over one-hundred thousand baud</i>
50	1	
75	2	
110	3	
134.58	4	
150	5	
300	6	<i>default</i>
600	7	
1200	8	<i>getting popular</i>
1800	9	
2400	10	
3600	11	
4800	12	
7200	13	
9600	14	
19200	15	

Now determine the number for how many bits in each word:

**Table 2: Bits**

<b>Bits</b>	<b>Number</b>
8	0
7	32
6	64
5	96

and add 16 (the 16 sets up the baud rate generator — it's not an option).

Next, determine the number for the desired stop bits:

**Table 3: Stop Bits**

<b>Stop bits</b>	<b>Number</b>
1	0
2	128

Choose your baud rate, bits per word (plus 16), and stop bits. Add their numbers together, then multiply by 256 and add 9 to the result of the multiplication (sounds like the start of a boring magic trick, doesn't it).

Now determine what parity you will use, and find the appropriate number:

**Table 4: Parity**

<b>Parity</b>	<b>Number</b>
none	0
odd	32
even	96
mark	160
space	224

Add this last number and you have the value for SE%.

**Example:** The default is 300 baud, which gives us a number 6 from Table 1. We are using an 8-bit word, so we add 0 (from Table 2), plus 16 to set up the baud rate generator:  $6 + 0 + 16 = 22$ . One stop bit adds zero (Table 3), so we multiply 22 by 256 and add 9, giving us

5641. *We are done since the default is no parity (Table 4).*

Unfortunately, we have one more step to do if the calculated value is greater than 32767. We must subtract 65536. For example, if we wanted two stop bits (128 from Table 3) we would have come out with  $((6 + 0 + 16 + 128) * 256) + 9 = 38409$ . In this case we would calculate  $38409 - 65536$  and would type, highlight and use the CALC command on the result:

SE%-27127

All this is much easier because we have the CALC command available to do the arithmetic.

***Designer's note:*** *We could have given you a long menu to run through to choose all these parameters, but the way we do it gives experts direct access to the registers of the Serial Card's 6551 chip. This would otherwise have required menus with choices like, "Do you want the mark parity bit transmitted and the parity check disabled?" and so on. Also, experts may note that if they add 13 instead of 9 to the product above, they will be able to send a break by setting SE%. To stop the break signal you will have to reset SE%. Fortunately, most systems no longer require breaks.*

### ***Garbled Electronic Communication***

If occasional characters are garbled, it is probably due to a poor phone connection. If you are getting total gibberish, it's possible that your setting of SE% doesn't match the other person's setting.

### ***Dealing With the Outside World — a Hint or Two***

If you are using a data service or bulletin board, it may have rules of its own that have nothing to do with SwyftWare, and this manual cannot help you there. Some of those external services have messages that will guide you through. We will only give one set of hints that we have found useful, hints which work most of the time: Starting with an empty text, first set LE%=29 as explained above (then delete the expression to keep your text clean). Now dial (by hand or with the modem) the telephone number of the service you want to use. We hope your modem has a speaker so that you can listen to the dialing, the ringing, and then hear two tones, one after the other. If information does not appear on your display within a few seconds, delete everything so that the text looks empty and use the SEND command two or three times. This often gets things started.

Lastly, once you get on, remember that you can leap to instructions and command tables that you have seen go by on your screen; you can even print them out. To save line charges, we recommend that you hold off on printing until after you sign off. Sometimes the cursor will stop blinking after a print. Just hold down the CONTROL key, and, while you hold it, press and release the RESET button and SwyftWare will be returned to normal operation, although you may no longer be connected. We therefore recommend again that you reserve any printing until after you are finished with your phone call.

***Personal note from the authors:** We've run into a lot of services and bulletin boards that confused us with their commands, so don't think that it's you if you have trouble. Many data services are very poorly designed, and nobody can figure them out without some help or a lot of experimentation.*

### **Control Codes**

Some bulletin boards or data services might ask you to use "control characters." For example, they might say, "To access the next message, type CONTROL Y." Sometimes they use a caret (^) instead of the word "control," so that the instruction might be "type ^Y". To do this with SwyftWare, press and hold the USE FRONT key, and, while you are holding it, tap first the letter "z" and then the letter asked for — a "y" in this example.

If you are asked to type an escape code (sometimes called ESC), press and hold the CONTROL (USE FRONT) key, and, while holding it, tap first the letter "z" and then the ESC key (labeled "PAGE" in the SwyftWare system).

You do not use the SEND command to send control characters. Use the technique we have just explained.

### **Receiving Information From the Modem**

When information comes into the SwyftWare system, it acts just as if another person were typing. All the things you can do with text that you create yourself can be done with text you get from the modem. On the other hand, text can become full when receiving just as when typing, and the same thing happens: the Apple bleeps. It also tries to tell the system sending information to stop.

We say "tries to tell" because not all systems use the same method.

But SwyftWare uses the most common technique, where a special code means to stop sending, and another special code means to start sending again. If incoming information does fill up the Text, do what you would do if typing had filled up the Text (see *Part 23, "Bleep When You Are Typing,"* and *Part 13, "When You Run Out of Room"*).

### ***Stopping Incoming Text and Restarting It***

Text coming from another SwyftWare-equipped Apple or from most data services can be interrupted by a CONTROL S. As explained above, press and hold CONTROL, and, while holding it, tap "z" and then "s".

To let the text continue, use CONTROL Q: press and hold CONTROL, and, while holding it, tap "z" and then "q".

This use of CONTROL S and CONTROL Q is sometimes called by the mysterious name *XON-XOFF protocol*.

### ***Ringling Your BEL: How to Wake Up the Person at the Other End of a Conversation***

You can try to make the terminal (or a SwyftWare-equipped Apple) at the other end bleep by sending a CONTROL G (appropriately known as the ASCII "BEL" code). Similarly, if SwyftCard receives a CONTROL G, it makes your Apple bleep. It is customary to send three or more bleeps for this purpose (hold CONTROL and type: zgzgzg).

### ***Using Your SwyftCard or SwyftDisk as a Message Center***

If your modem can answer the phone (auto-answer) then, providing you leave your system turned on and the modem hooked up, people can leave messages for you by sending text from their system. However, they cannot operate your computer remotely through SwyftWare or get any information out of it — it is a very secure system.

You should tell anyone who might send you a message that unless they send a CONTROL E to your SwyftWare they will get no reply, nor will they have any way of confirming that your system is working. When they do send a CONTROL E, your SwyftWare will automatically send back

SwyftCard 1131

(The number may be different if they have a different version of SwyftWare). This way they know that your system is operating correctly.

### ***Other Uses of SEND***

The SEND command may be used with serial devices other than modems. For example, by attaching a speech synthesizer (such as an Echo GP\* or Votrax Type'n Talk\*) to the Apple Super Serial Card, you can have the messages that you send spoken for you.

*\*Echo GP is a trademark of Street Electronics; Votrax and Type'n Talk are trademarks of Votrax Inc.*

# **Chapter V**

## ***Problem Solving & Information Retrieval***

<i>Part 21</i>	<i>Stopping</i>	<i>89</i>
<i>Part 22</i>	<i>Emergency Restart</i>	<i>91</i>
<i>Part 23</i>	<i>Bleeps</i>	<i>93</i>
<i>Part 24</i>	<i>Information Retrieval and Other Applications — a Few Ideas</i>	<i>98</i>



## **Part 21: Stopping**

### **To Stop Printing**

Press and hold the USE FRONT key, and, while you hold it, press and release the RESET key. The unprinted text will remain highlighted. Use the PRINT command again to begin printing from the point you left off.

### **To Stop a BASIC Program**

Press and hold the USE FRONT (CONTROL) key, and, while you hold it, press and release the RESET key. The program will halt and the text generated by the program will be highlighted. If you now use the CALC command, the program will begin to run again *from the beginning*. You can also use CONTROL C as explained in Apple's manuals.

Certain errors in the typing of calculations, printout format changes, or printer initialization strings can cause the Apple to bleep. An on-screen message, "?SYNTAX ERROR", pops out when you make such a mistake. If you accidentally use the CALC command on a large chunk of highlighted text, the computer may start hiccuping:

?SYNTAX ERROR	(bleep!)

CONTROL-RESET will stop the hiccups, or you can wait for them to stop by themselves, which they will do eventually.

### **Stopping an Incoming SEND**

When communicating with another SwyftWare-equipped Apple or with most data services, the following techniques will stop an incoming send:

It is usually possible to stop an incoming transmission temporarily by holding down the CONTROL key and pressing and holding the Z key for as long as you want it to stop.

To stop a transmission temporarily without having to continuously hold down a key, hold down the CONTROL key, and, while you hold it, tap first "z" and then "s". (To restart a transmission that you have stopped, hold the CONTROL key and tap first "z" and then "q"). This is called the XON-XOFF protocol.

If XON-XOFF does not work for you, refer to the manual for the system you are communicating with. The manual should show you how to stop or interrupt a message.

### ***Stopping an Outgoing SEND***

Use CONTROL-RESET.

***Note:*** *This will stop the outgoing SEND, but it will cause your modem to drop its carrier, usually causing the other person's modem to hang up.*

## **Part 22: Emergency Restart**

### **Frozen Cursor**

If the cursor stops blinking and does not respond when you type, press and hold the CONTROL (USE FRONT) key, and, while you hold it, press and release the RESET button at the upper right corner of the keyboard. You will *not* lose any text. The cursor freezes during the time a BASIC program is running. An error in a BASIC program, certain BASIC statements, or starting a SEND or PRINT when the modem or printer is turned off can also cause the cursor to freeze.

When the text is full, the cursor stops moving but continues to flash. This is not the same as a frozen cursor, which neither moves nor flashes. When the text is full, the Apple will bleep every time you tap a key (see Part 23, "Bleep When You Are Typing," for the remedy).

### **Overriding the DISK Command**

Normally we trust the DISK command to save text or load a new disk as needed. You can sidestep the DISK command and its safety features if you wish, but not without risk: if you haven't saved the text on-screen, you may lose valuable changes when you force SwyftWare to load the disk in the drive. We call this procedure *restarting*. You should avoid using it. If you begin to rely on this technique instead of the DISK command, eventually you will clobber yourself with it — that is to say, you'll accidentally lose some valuable text. A better procedure is described under "Force Loading" in Part 13.

Restarting consists of starting the disk drive and loading whatever information is on the disk. Here's how to do it: Press and hold the CONTROL key and Open-Apple, and, while you hold both these keys, press and release the RESET button at the upper right corner of the keyboard. Then release CONTROL and Open-Apple. The drive whirs, the on-screen text vanishes, and the computer loads whatever disk you have in the drive (SwyftWare or not).

You can also override the DISK command by turning the power to your Apple off and then back on.

**SwyftCard users:** *If there is no disk in the drive, you must turn the power to the computer off and back on to restart. You will then see page characters 0 and 1, the cursor and highlight, and that's all*

*(this screen is illustrated at the beginning of Part 6). Obviously, you can't restart SwyftDisk without a disk in the drive because the system is based on the disk.*

If there is no disk in the drive when you use CONTROL-OPEN-APPLE-RESET, the drive will spin forever while the screen displays "Apple II" or Apple //e". If you press CONTROL-RESET, the drive will stop and you will be in BASIC. If you put a SwyftCard disk in the drive while the drive is spinning and then close the drive door, the disk will load.

**WARNING:** *Whenever you override the DISK command, the on-screen text is lost forever. If you don't intend to lose the on-screen text, make very sure that you first have the text safely stored on a disk.*

## ***Part 23: Bleeps***

A *bleep* is an audible signal from the Apple. It signifies a refusal or inability to carry out a command, or it is a warning.

### ***Bleep When You Are Typing***

1. The most common reason by far is that you have run out of room. This can happen when typing characters or when tabbing (which is the same as typing spaces). When you are out of room you get a bleep on every keystroke and the cursor won't budge even though you are tapping keys (it will leap, however).
2. An attempt to add a page character to the text when there are more than 200 page breaks or page characters already present will cause a bleep.

If the problem is lack of room, you can sometimes get more by highlighting a couple of characters and deleting them. Remember that SwyftWare saves the last chunk of text you deleted, and this saved chunk consumes space. A big chunk can consume a lot of space. When you highlight and delete two characters, SwyftWare throws out that space-consuming chunk. You get the space back, less room for two characters (the chunk you just deleted).

In general, the best thing to do is to start another disk. If what you are working on is just part of a much bigger piece, or you need room to edit the rest of the text, use copying up to transfer part of the Text to a fresh disk (this procedure is explained in *Part 13, "When You Run Out of Room"*).

### ***Bleep When You Use the INSERT Command***

When you INSERT a chunk of text and the system bleeps, this warns you that there is no longer room to INSERT the same chunk again. In fact, even if you deleted enough text character-by-character so that the chunk last INSERTed would seem to be able to fit, you will find that nothing happens if you try to INSERT it again (but nothing was lost — you INSERTed the chunk into your text, so it is available for further use).

If you use INSERT to make additional copies, you are creating additional text. This is the same as if you had typed the new material, and you may run out of room. See *Bleep When You Are Typing* on the previous page.

### ***Bleep When You Use the DISK Command***

1. There is no diskette in the drive. Oops — put one in.
2. The drive door is open. Close the door and try DISK again.
3. You've put a new disk in the drive without saving the changes in the Text on-screen. If SwyftWare does a save, you will lose whatever is on the new disk. If it brings in the Text from the new disk, then you will lose your changes. SwyftCard doesn't want to lose any text no matter what, so it bleeps in frustration.

You can do one of two things: a) Put back the disk that belongs with the on-screen text and use the DISK command to save the changes, then put in the new disk and use the DISK command to bring it on-screen. Or, b) if you are absolutely certain you don't care about saving your changes, put the new disk you want to load in the drive, then DELETE everything (leaving only page characters 0 and 1 on the screen), and use the DISK command.

4. The disk is write-protected and SwyftWare tried to do a save. Take out the disk and remove the write-protect tab.
5. The disk or the drive is damaged.

### ***Bleep When You Copy Up***

If there isn't enough room in the receiving Text for the material you are trying to copy up, SwyftWare will bleep. The fix is to copy up less, or to make more room by deleting some information from the receiving Text.

### ***Bleep When You Use the CALC Command***

1. A calculation can generate too many characters to fit in the computer, just as typing can. Running a BASIC program with the CALC command can do the same thing. In either case, SwyftWare will bleep. Split the Text onto two disks, or make room by deleting part of the text.

2. Certain errors in the typing of calculations, printout format changes, or printer initialization strings can cause the Apple to bleep. An on-screen message, such as "?SYNTAX ERROR", also pops out when you make such a mistake. The problem is often a simple typing error. Make certain you have typed the expression exactly as it is written in this manual, then start over.

This bleep can also occur if you press the CALC command by mistake — when you intended to PRINT or SEND, for example.

### ***When You Use the SEND Command***

1. If the person sending you text sends more than SwyftWare's capacity to receive and ignores SwyftWare's request to stop sending, you get a bleep. Every character that overflows the limit will cause a bleep, and since the characters are sent at a high rate of speed by the modem, the beeping sounds something like an alarm clock going off.

The beeping will continue as long as characters are being sent and will stop once the transmission is over. It can also be stopped by unplugging the phone, which breaks the connection.

2. If the sender sends a CONTROL G, the Apple will bleep. The method of sending a CONTROL G from SwyftWare is to hold down the CONTROL key and tap first "z" and then "g".

### ***Asking SwyftWare Why You Got a Bleep***

Bleeps are very rare with SwyftWare, and it is usually immediately apparent what caused it and how to fix it. This is why we don't have the usual and bothersome error messages that abound in most computer products. However, deep inside, our system knows just what caused each bleep, and while we don't use this facility ourselves much, we thought we'd let you know how to find out what SwyftWare thinks caused the bleeps. Just type

```
?PEEK(5630)
```

highlight this expression, and use the CALC command. A number will appear. Find the number in the "Bleep Table" that follows to see what SwyftWare thinks is the problem:

## **Bleep Table**

### **No.    *Diagnosis***

- |     |  |
|-----|--|
| 201 | No error   |
| 1   | You are trying to save onto the wrong disk. The disk in the drive does not correspond to the Text on-screen.   |
| 2   | You saved the on-screen Text, then moved the cursor or typed something before you tried to save onto a backup disk. If SwyftWare allowed this, the backup would be different than the original. Put the first disk back in the drive, use the DISK command, then use the DISK command immediately on the backup. |
| 3   | The disk is write-protected. If you get this bleep and the disk is definitely <i>not</i> write-protected, the write-protect switch in your disk drive is broken.   |
| 4   | You're trying to load a non-SwyftWare disk before saving the SwyftWare Text now in the computer.   |
| 5   | The text you're trying to copy up will not fit into the Text you are trying to load.   |
| 6   | The text you're trying to copy up causes too many page breaks in the Text you are trying to load. Start over with a smaller highlighted chunk or delete material from the Text you want to load.   |
| 7   | The GET command failed (this cannot happen unless you have installed the GET command; see <i>Appendix D</i> ).   |
| 10  | You're typing when the text is full or when there are too many page breaks.  |
| 11  | You're using TAB when the text is full.  |
| 12  | There is not enough room to INSERT the same chunk of text again.   |
| 13  | There is nothing to INSERT.  |

- 14 Text is coming in over the modem when Text in the computer is already full or has too many page breaks.
- 15 Using the CALC command caused full Text or too many page breaks.
- 16 Using the CALC command caused a BASIC error.
- 17 Typing characters caused too many page breaks.
- 18 Using TAB caused too many page breaks.
- 19 Using the INSERT or GET command, or changing WI%, PA%, PL%, AB%, or BE% caused too many page breaks.
- 30 The disk drive door is open, the disk is bad, or there is a hardware flaw in the disk drive or interface.
- 31 SwyftWare couldn't read the text on one track of the disk *(for experts only: track number didn't match)*.
- 32 SwyftWare couldn't read the text on one track of the disk *(for experts only: checksum didn't match)*.

**Note:** An error message remains in memory until the next error message replaces it. Saving a Text also saves the current error message.

## Part 24: Information Retrieval and Other Applications — a Few Ideas

*Information retrieval* is a ten-dollar term that means nothing more than being able to find what you want after you've tucked it away somewhere. The technique of leaping is not only a means of moving the cursor, but of moving the cursor to a particular item in the Text — hence leaping *is* information retrieval.

Suppose you have a text consisting of people's names and telephone numbers, like this:

678/526-1234	Harold DaSilva, 34 Dusty Rd.	San Palinicia, CA
777/352-9666	Theodore Chiesa, 14900 Eightfold Way	Sinkerville, TX
292/446-2956	Peter's Pizza, 12 Della St.	Oleeklee, OK
892/501-1841	Patricia Lewis, 1200 Appellate Ct.	Tavistock, CT
292/101-8632	Livestock Protective Assoc.	Two Hats, OK
	Avery Thixill, pres. 292/102-2342	
	M. Violet Blanc, treasurer 292/446-1202	
	Michael Loukovich, cowhand	
498/184-3338	Fandango Frank, 5622 Dress Circle	Wobston, MA
	Manuel Crozier Heintz, 82 Camino Cerrado	
292/615-9235	Kim Mascoli, 11206 Lois Ln.	Mantis, OK
555-1252	Doctor Fenton Farnsworth	
398/783-8114	David Barnett, 6266 Warp Drive	Stony Shoe, MA
292/152-6945	Calixto Skelton, 626 Wrong Way	Spoon River, OK

and so on (these numbers and names were made up, if you haven't guessed this already). Say that you keep such a phone list in the Text you are currently using. Whenever you want to look up somebody's number, all you have to do is leap to their name. Notice that you can reference (leap to) their first name or last name, their street address or street name, even their town or state. With at most a few taps of the LEAP AGAIN key, you are sure to find them.

You can use such a list in other ways, too. For example, to find all the people you know in Oklahoma, leap to "OK" and use the LEAP AGAIN key. To find your doctor's phone number, just leap to "doctor".

Let's say that you had a few letters in the same Text as your phone numbers. You could find the letters by leaping to "Dear", and with each

press of the LEAP AGAIN key you would be able to look at another letter. It won't take more than a second or two to find the letter you're looking for.

Even faster — at least most of the time — is looking for a key word inside the letter. If you want to see the letter you wrote a while ago about a new brand of hula hoops you discovered in a local store, you could leap to "hula" or "hoop", and there would be the letter you wanted.

Similarly, if you had typed in a lot of recipes, you could not only look up a recipe for, say, *chile con carne* by leaping to the name of the recipe, but if you had picked a dozen zucchinis in your garden, you could search for all recipes with zucchini in them just by leaping to "zucchini" and using the LEAP AGAIN key.

Clever arrangement of items can make certain kinds of retrieval easier. Take the phone list, for example. A leap to "Livestock Protective Association" will also give you the names of all the officers. Similarly, a leap to any of the people who work for the Association will turn up the Association phone number, plus the names of all the other people in the group. Notice that with SwyftWare information does not have to be complete. Not all names in the list have addresses or phone numbers, nor do they need to. Can you only remember a first name or part of a last name? SwyftWare doesn't care. In fact, you don't even have to line up all the names and numbers as we've done here. They can be spread helter-skelter all over your text and you can still find them.

SwyftWare can be used to do inventories, mailing lists, and many other applications, all without programming or learning anything beyond typing, leaping, and elementary editing.

## ***"Mail Merge" Applications***

If you want to send the same letter to a number of people, you might start by arranging their names and addresses like this:

```
Paul Conroy
21 Easy St.
Soy, NY 12534
Dear Paul,
\
Andrew Johnson
1131 Swyft Wy.
Sundapp, ND 73453
Dear Andrew,
\
```

and so on. Then you would type the letter, highlight it, and delete it. Next you'd leap to a backslash (\), delete the backslash, and use the INSERT command. This inserts the letter under the address and greeting. Highlight and print the letter, then do the same thing for each address. It's a lot easier than doing it by hand. If you had to do it for a hundred letters every week, this would be too tedious, and you'd probably be well-advised to get a conventional "mail merge" program. The SwyftWare  $\longleftrightarrow$  ProDOS converter might be used to move your address list to a ProDOS file for this purpose.

We'd like to show one more application that uses BASIC and information retrieval. Say that you'd like to make up a calendar for April — which hath 30 days. If you know BASIC, you might write a little program, like this:

```
10 PRINT
20 FOR I = 1 TO 30
30 PRINT "APRIL "; I
40 NEXT I
RUN
```

highlight it, and use the CALC command. You will get:

APRIL 1  
APRIL 2  
APRIL 3  
APRIL 4

and so on, up to

APRIL 30

And now, by leaping to any given date, you can make entries in your calendar or find out what's happening on a date you've already filled in. For example, you could find out a friend's birthday (assuming you've put birthdays on the calendar) merely by leaping to your friend's name. You could find all birthdays by leaping to the word "birthday" and using the LEAP AGAIN key.

With experience you will find more and better uses for SwyftWare. The wealth of applications is endless.



## ***Appendices***

<i>Appendix A</i>	<i>ProDOS Converter</i>	102
<i>Appendix B</i>	<i>Theory of Operation</i>	111
	<i>Schematic</i>	117
<i>Appendix C</i>	<i>List of Features</i>	118
<i>Appendix D</i>	<i>Reassigning the Key Commands</i>	121



## ***Appendix A: ProDOS Converter***

The ProDOS Converter program is on the flip side of the SwyftWare tutorial disk, which is included in your SwyftWare pack. This program enables you to move SwyftWare text into ProDOS files and vice versa.

### ***Equipment***

You can do file conversion if you have one of the following:

- Apple //e with two disk drives
- Apple //e with a single disk drive and an extended 80-column card (or other memory addition)
- Apple //c

You cannot do file conversion with only one disk drive and 64K of memory (regular 80-column card). Most ProDOS users find two drives necessary anyway. Of course, an extended 80-column card costs less than a second drive.

### ***How to Boot the ProDOS Converter Program***

Don't do this now — wait until you come to this step in your file transfer procedure. We're explaining it now so we don't have to repeat this information in several different places.

*Boot* is computer jargon for "start up." To boot your ProDOS Converter program, turn the tutorial disk over (unlabeled side up) and place it in Drive 1 (the drive whose red in-use light comes on when you turn on the power to your computer).

- If the computer is off, the ProDOS Converter will boot when you turn the system on.
- If you are already in the SwyftWare system, simply use the DISK command to boot the ProDOS Converter program.
- If you are running other Apple //e software, reboot the system using the proper procedure for the program you are running.

After a few seconds you will see the SwyftWare ProDOS Converter main menu:

```
*****  
* SWYFTWARE <--> PRODOS CONVERSION *  
* Version 1.21 Copyright 1985 by *  
* Information Appliance Inc. *  
*****
```

- 1) CONVERT SWYFTWARE TO PRODOS
  - 2) CONVERT PRODOS TO SWYFTWARE
  - 3) EXIT TO BASIC
  - 4) EXIT TO SWYFTCARD
- SELECT AN OPTION

### ***How to Use the Menu***

Select the option you need by typing a number from 1 to 4. Simply typing the number will start the program (you don't have to tap the RETURN key, too). You can get back to the main menu any time by pressing the ESC key (labeled ESC/PAGE in SwyftWare system) when the program asks for an input.

### ***Choose Your File Transfer Procedure***

How you transfer files depends on your equipment and whether you are moving a ProDOS file to SwyftWare or the other way around. Find the heading below that describes what you want to do and follow the step-by-step recipe.

*Source disk* means the disk that contains the text you want to transfer.

### ***Two Disk Drives: ProDOS → SwyftWare***

**Materials:** Two disk drives, SwyftWare ProDOS Converter disk, ProDOS source disk (know the name of your ProDOS disk and ProDOS file), blank disk to receive ProDOS text.

1. Boot the SwyftWare ProDOS Converter in Drive 1.
2. Main menu appears. Select "2 - CONVERT PRODOS TO SWYFTWARE".

3. Menu says, "INSERT BLANK DISK IN DRIVE 1." Remove the ProDOS Converter and place your blank disk in Drive 1.

**CAUTION:** *The converter program does not look to see if the disk in Drive 1 is really blank. If the disk already has text on it — SwyftWare or ProDOS — that text will be wiped out when the ProDOS file is copied. Be sure the disk you are using in Drive 1 does not contain anything valuable!*

Press RETURN. Drive whirs, then screen says

```
ENTER PATHNAME  
?
```

4. Put your ProDOS source disk in Drive 2.
5. Type (all capitals)

```
/DISKNAME/FILENAME
```

*DISKNAME* is the name of your ProDOS disk. *FILENAME* is the name of your ProDOS file on that disk. (For example, if your disk was named DISK1 and your file was named JUNK, you would type "/DISK1/JUNK".)

Press RETURN. The disk drives whirl. When they stop, the ProDOS file has been copied in an intermediate form to the disk in Drive 1.

6. New menu appears. Select "4 - EXIT TO SWYFTCARD".

Drive 1 whirs and an empty SwyftWare text appears on the screen (though the drive was activated, the text did not load).

7. Use the DISK command to load the newly transferred text. It will be highlighted when it comes on-screen.
8. Type

```
CALL 3600
```

highlight it, and use the CALC command. This erases the disk in Drive 1. Tap the DELETE key to erase "CALL 3600".

9. Use the DISK command. This saves the text in SwyftWare form on the disk in Drive 1.

### **Two Disk Drives: SwyftWare → ProDOS**

*Materials: Two disk drives, SwyftWare ProDOS Converter disk, SwyftWare source disk, formatted ProDOS disk (know the name of your ProDOS disk and ProDOS file).*

1. Highlight the SwyftWare text you want to transfer. Use the DISK command to save the text with the highlight extended.
2. Remove SwyftWare disk, insert SwyftWare ProDOS Converter disk in Drive 1 and boot with DISK command.
3. Main menu appears. Select "1 - CONVERT SWYFTWARE TO PRODOS".
4. Menu says, "INSERT SWYFTWARE DISK IN DRIVE 1". Remove the Converter disk from Drive 1 and insert the SwyftWare disk with your highlighted text saved on it.

Tap the RETURN key.

Menu now says "ENTER PATHNAME"  
?

5. Put formatted ProDOS disk in Drive 2.
6. Type

`/DISKNAME/FILENAME`

where *DISKNAME* is the name of the ProDOS disk, and *FILENAME* is the name of the file receiving your SwyftWare text.

Tap the RETURN key.

The disk drives whirl, as your SwyftWare file is copied to the ProDOS disk in Drive 2.

**Note:** *If the ProDOS file you name does not exist, it will be created. If it already exists, the SwyftWare text will replace the contents of the file.*

7. When copying is complete, the main menu reappears. Select whatever you want to work on. If you wish to work on the new ProDOS file, load your ProDOS software, then load the file as directed by the software.

### **One Disk Drive: ProDOS → SwyftWare**

*Materials: One disk drive, at least 128K RAM (Apple //c or //e with extended 80-column card or other memory enhancement), ProDOS Converter disk, ProDOS source disk (know the name of the ProDOS disk and ProDOS file), blank disk to receive ProDOS text.*

1. Boot SwyftWare ProDOS Converter.
2. Main menu appears. Select "3 - EXIT TO BASIC".
3. BASIC prompt ( ) appears. Type

`-FILER`

Press RETURN. Disk drive whirs as Apple program called "FILER" loads.

4. Menu appears. Select "F - FILE COMMANDS".
5. Menu appears. Select "C - COPY FILES".

"Copy Files" screen appears. The cursor is next to

`PATHNAME: (`

on the screen.

6. Remove the SwyftWare ProDOS Converter disk. Insert ProDOS source disk.
7. Type

`/DISKNAME/FILENAME`

where *DISKNAME* is the name of the source disk and *FILENAME* is the name of the file you want to transfer to SwyftWare. (For example, if your disk was named DISK1 and your file was named JUNK, you would type "/DISK1/JUNK".)

Press RETURN.

The cursor moves down, to the right of parentheses in

TO PATHNAME: (

8. Type

/RAM/FILENAME

where *FILENAME* is the name of the file you want to transfer.

Press RETURN.

9. Menu displays

-- INSERT DISKS AND PRESS <RET> --

Do not insert any disks — simply press RETURN.

Disk drive whirs until menu displays "COPY COMPLETED". Your ProDOS file is now in RAM.

10. Tap the ESCAPE key (labeled ESC/PAGE in SwyftWare system). "FILE COMMANDS" menu appears.

11. Tap the ESCAPE key. Main menu appears.

12. Tap the Q key. "QUIT" screen appears.

13. Remove your ProDOS source disk. Insert the SwyftWare ProDOS Converter disk in the drive.

Press RETURN. Disk drive clacks and whirs.

14. Main menu for ProDOS Converter appears. Select "2 - CONVERT PRODOS TO SWYFTCARD".

15. Menu says, "INSERT BLANK DISK IN DRIVE 1". Remove the ProDOS Converter and insert the disk on which you want to copy the file now in RAM.

**CAUTION:** *The Converter program does not look to see if the disk in Drive 1 is really blank. If the disk already has text on it — SwyftWare or ProDOS — that text will be wiped out when the ProDOS file in RAM is copied. Be sure the "blank" disk does not contain anything valuable!*

Press RETURN.

Menu says, "ENTER PATHNAME".

16. Type

```
/RAM/FILENAME
```

where *FILENAME* is the name of the file now in RAM.

Press Return. Drive whirs as ProDOS file copies to the disk.

17. Main menu appears when copy is completed. Select "4 - EXIT TO SWYFTCARD".

Drive whirs and empty text appears on screen (though drive was activated, the text did not load).

18. Use the DISK command to load the newly transferred text. It will be highlighted when it comes on-screen.

19. Type

```
CALL 3600
```

highlight it, and use the CALC command. This erases the disk. Tap the DELETE key to erase "CALL 3600".

20. Use the DISK command. This saves the text now on-screen.

### ***One Disk Drive: SwyftWare → ProDOS***

*Materials: One disk drive, at least 128K RAM (Apple //c or Apple //e with extended 80-column card or other memory enhancement), ProDOS Converter disk, SwyftWare source disk, formatted ProDOS disk (know the name of the ProDOS disk and ProDOS file).*

1. Highlight the SwyftWare text you want to transfer. Use the DISK command to save the text with the highlight extended.
2. Remove SwyftWare disk, insert SwyftWare ProDOS Converter disk in drive and boot with DISK command.
3. Main menu appears. Select "1 - CONVERT SWYFTWARE TO PRODOS".

4. Menu says, "INSERT SWYFTWARE DISK IN DRIVE 1". Remove the Converter disk and insert the SwyftWare disk with your highlighted text saved on it.

Tap the RETURN key.

Menu now says "? ENTER PATHNAME"

5. Type

`/RAM/FILENAME`

"RAM" tells the computer you are using RAM to make the copy. *FILENAME* is the name of your new ProDOS file.

Press RETURN.

The drive whirs as the SwyftWare file is copied into RAM.

6. Menu appears when copy is complete. Select "3 - EXIT TO BASIC". A BASIC prompt ( ) appears.
7. Remove the SwyftWare source disk. Insert the SwyftWare ProDOS Converter disk.

8. Type

`-FILER`

Press RETURN.

Disk drive whirs as Apple program called "FILER" loads.

9. Menu appears. Select "F - FILE COMMANDS".
10. New menu appears. Select "C - COPY FILES".
11. Conversion screen appears, with cursor next to

PATHNAME: (

Type

`/RAM/FILENAME`

where *FILENAME* is the name of your new ProDOS file. Press RETURN.

12. Cursor moves down the screen next to

TO PATHNAME: (

Type

/DISKNAME/FILENAME

where *DISKNAME* is the name of your ProDOS disk and *FILENAME* is the name of your ProDOS file. Press RETURN.

13. Menu says, "INSERT DISKS AND PRESS RETURN". Remove the ProDOS Converter disk from the drive and insert your *formatted* ProDOS disk.

Press RETURN.

The disk drive whirs. When the screen says "COPY COMPLETED" your copy is saved on the ProDOS disk.

*Note: If the ProDOS file you name does not exist, it will be created. If it already exists, the SwyftWare text will replace the contents of the file.*

### ***Converting AppleWorks and DOS 3.3***

To convert AppleWorks or DOS 3.3 files to SwyftWare, first convert them to a ProDOS text file. See Apple's manuals. Apple's ProDOS Filer and ProDOS DOS 3.3 converter programs are included on the SwyftWare ProDOS Converter disk.

### ***Oversized Texts and Other Errors***

If the SwyftWare text to be converted is too long for the resulting file to fit on the volume specified in the pathname, an error message will appear and as much of the text as will fit in the ProDOS file will be transferred. If the ProDOS file you name does not exist, it will be created. If it already exists, the SwyftWare text will replace the contents of the file. If the disk is write-protected or the ProDOS file is too large for the SwyftWare (more than 40 Kbytes) an error message will appear. Tap the ESCAPE (PAGE) key to return to the main menu, and repeat the process with a non-write-protected disk, or with a smaller ProDOS file.

## ***Appendix B: Theory of Operation / Schematic***

This Appendix is divided into three sections: interface, software, and hardware. The sections are not system documentation, but guides to the spirit of design in each area.

### ***SwyftWare User Interface Theory of Operation***

The paradigms used in SwyftWare were invented to cure a host of problems shared by almost all current systems — most of them small enough in their own right, but which taken together make learning and using conventional software far more time-consuming than necessary, and which make using computers a frustrating and annoying process.

We have always wondered why, for example, you have to format disks — isn't the computer smart enough to see if a disk isn't formatted and do it if necessary? We find cursor control keys far too slow, and when you consider the number of auxiliary commands they require (move to next/previous word, sentence, paragraph, page, move to beginning/end of line, document, file...) we find them too complex. The mouse is only a small improvement since it takes your hands away from the keyboard, and uses up much screen space for menus, scroll bars, and the rest of the associated mouse apparatus. We are annoyed when we are put through menus instead of being able to do what we want right now, and we are puzzled by the huge number of commands in most systems. We hate disk systems that allow you to lose work through trivial human error. We are amazed that many word processors cannot keep up with human typing speed.

SwyftWare shows that with proper design all these questions and bothers — and many others that have plagued us for years — can be answered and fixed. And the system works on an inexpensive computer with only one disk drive, with minimal memory requirements. Our product does what most people need done — without an operating system, expensive price tag, or bells and whistles.

The major design principles include numerous innovations, as well as applications of what we have learned from the work of others.

1. The cursor LEAP concept, whose average time to target is about three times faster than that of the most advanced method in common use up until now: the mouse.
2. The cursor itself, whose two parts show you exactly where what you type will appear and where delete will operate. The cursor also collapses upon being moved so that you do not have to aim "one off" if you want to delete.
3. A very small set of fundamental operations that allows you to accomplish a wide range of tasks with ease.
4. The elimination of an operating system, thereby allowing all operations to be performed directly and immediately from the editor without having to go into different modes.
5. The elimination of modes in general, which makes habit formation easy since you do not have to think about what state the system is in to figure out what a command will do. This property is called *modelessness*.
6. Not providing many ways to do a task — again so that you do not have to think about alternate strategies when you are about to do something. We call this principle *monotony*. Like modelessness, monotony aids habit formation.
7. The emphasis on habit formation is itself a fundamental principle of the design, and one often overlooked by other designers. We consider it important that after a brief period of learning, a user should not have to think about the system while using it.
8. The DISK command, which simplifies the usual complexities of a DOS (Disk Operating System) into one simple command. It also provides protection against most common mistakes that would cause a loss of data on other systems. The technique of making one disk correspond to one Text is what makes this command possible.
9. The emphasis on speed of operation proportional to frequency of use (often-done tasks must be very fast, seldom-done tasks can be slow).
10. What you see is what you get — the way it looks on-screen is the way it prints on paper. (This principle was violated for underlining

due to a limitation in Apple display hardware.)

11. Noun-verb design of commands. First you specify what you are going to work on (which gives you time to make sure you are right and to make corrections), *then* you give the order as to what to do. Some systems work the other way around, or even worse, mix the two styles.
12. It is very hard to louse yourself up or clobber something you are working on. It's not impossible, but it's hard enough to do that it is not likely to happen by chance or a momentary lapse of attention.
13. The inclusion of programming and communications within a general purpose environment, where the output is placed in the editor/retrieval environment.
14. The allowance of months of testing and reworking time in the schedule, so that purchasers of the system are not being used as test subjects.

This is only a barest sketch — the system specs run to some 50 pages — but we hope it gives you a feel for what led us to design SwyftWare the way we did.

### ***SwyftWare Software Theory of Operation***

The system is small and operates quickly partly because it is implemented in FORTH and assembly language, and partly because it has an inherently clean and simple design. There are few commands, and they operate uniformly. Text is not cluttered with special markers. All of this minimizes programming effort.

The structure of the text, although not unique to SwyftWare, is key. The beginning of the text (from the first character up to the last character in the highlight) is stored in the lower portion of the text area, and the end of the text (from the character at the cursor or one after the cursor to the last character of the text) is stored in the upper end of the text area. Between the upper and lower texts lies the gap. This means that typing just puts characters into the gap, and thus can proceed very rapidly. Updating the screen is the only task that need be done while typing is going on, which is one reason why it is possible for SwyftWare to do word wrap and unwrap on the fly. The other reason is the uniformity of text.

Searching is fast because text lies entirely in RAM, contains no codes or other obstructions, and is in only two contiguous areas. The disk operates quickly because we do not use Apple's encoding scheme, and because we only write or read as much text as necessary. As a consequence of this decision, a ProDOS conversion routine is provided to establish a link to other Apple software. Formatting is done on the fly, since we write a whole track at a time, including sync bits. Not only is this fast, but it eliminates the need for a separate formatting step on the part of the user.

The system pointers are stored on disk so that SwyftWare texts come up in the exact state they were last saved. A serial number is written on each disk so that we can detect whether it is the same disk that was booted or if the user has changed disks. When backup disks are made, the same serial number is written on the master and all backups.

Updating the display after a leap is sped up by having a table of pointers to the places where pages begin. Thus, in order to figure out how the text should be formatted, the display algorithm has only to go back to the nearest page break prior to the text that is to be displayed at the beginning of the screen.

When inserting text, large areas of memory may have to be moved bodily. This "brute force" approach is surprisingly fast, but with very large texts does lead to a perceptible slowing. Still, SwyftWare is much faster than any other program that does a similar task on the Apple.

A deletion puts the deleted text at the beginning of the text area and moves the lower part of text up out of the way. This means that there is no limit to the amount of text that can be deleted, since we do not have to set aside room for a separate delete buffer.

Decisions of this sort abound, ultimately leaving an unusually large amount of space for the user — in this case over 80% of the memory that Apple does not dedicate to specific uses. SwyftWare does not use the extra memory afforded by the extended 80-column card, as the bank switching required would make the program operate too slowly for the high-quality interaction we think important.

The CALC, PRINT, and SEND command are all fundamentally the same. They take the highlighted chunk of text and transmit it to the BASIC interpreter, the printer port, and the serial port respectively.

Of these, dealing with the BASIC interpreter is the most difficult, since SwyftWare operates in the same address space as BASIC.

The keyboard tables are in RAM so that software developers can add new commands or redirect existing commands to execute code that the developers provide. BASIC programs that amplify SwyftWare's abilities can also be written. By changing the value of one variable (BT%), the bottom of text can be moved up so that developers can have room for their own code.

SwyftWare's software amounts to less than 16K bytes of code. Approximately half is in a tokenized FORTH, and half in assembly code.

### ***SwyftCard Hardware Theory of Operation***

The SwyftCard is a plug-in card for the Apple //e and operates in Slot 3. The card contains three integrated circuits that provide a power-on reset circuit, storage for the SwyftCard program, and control signals for the card. The card operates by asserting the Apple //e bus signal INH', which disables the built-in ROM and enables the SwyftCard ROM. This permits the SwyftCard program to take over the system at power-on and run the SwyftCard program. (Please refer to the schematic.)

The LM311 voltage comparator is connected to provide the power-on reset function. When the Apple //e is first turned on, the power-on reset circuit resets the PAL, turning on the SwyftCard and disabling the Apple //e internal ROM. The power-on reset circuit must be provided because the existing Apple //e reset function is used by many Apple //e programs for a "warm start": if Apple //e reset always started the SwyftCard, other programs could not use the warm start.

The 27128 PROM is used to store the SwyftCard program. The PROM contains 16384 bytes which are mapped into the address space \$D000 - \$FFFF. Since the address space is only 12K, there are two 4K sections of the PROM mapped into the address space \$D000-\$DFFF.

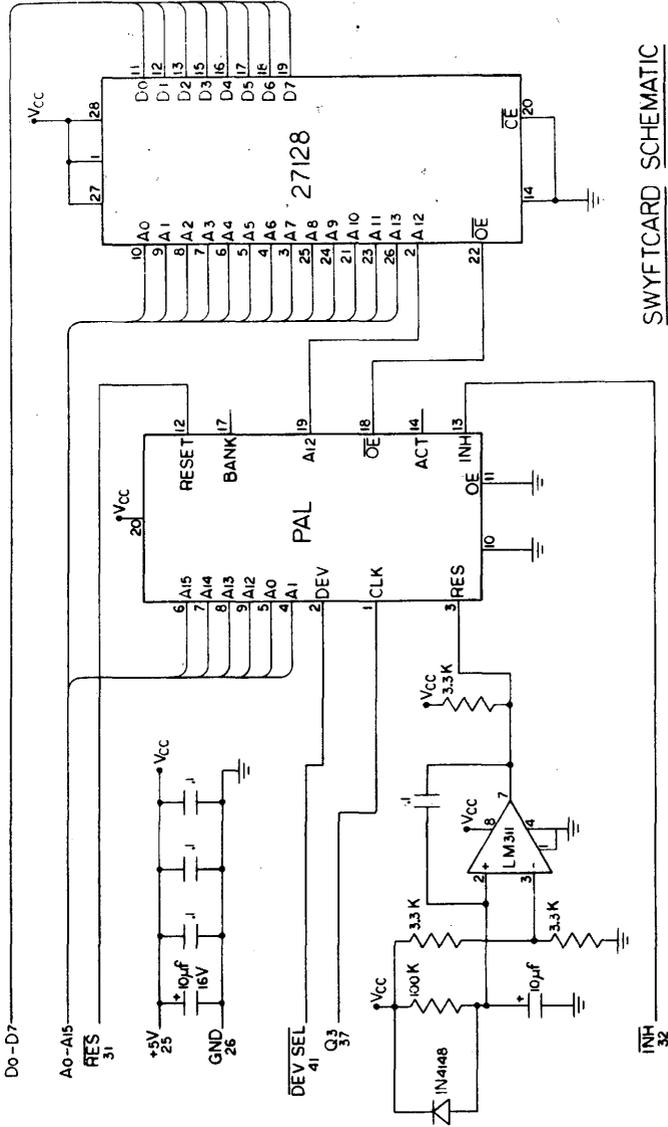
The card is controlled by the PAL. When the SwyftCard is active, the PAL asserts the INH' signal, enables the PROM, and bank switches the \$D000-\$DFFF address space. The card is controlled by two soft switches. The soft switches are controlled by accessing the following memory locations with either a read or a write operation.

\$C0B0 - SwyftCard active, Bank 1  
\$C0B1 - SwyftCard inactive, Bank 1  
\$C0B2 - SwyftCard active, Bank 2

When the power-on reset circuit asserts the RES signal on Pin 3 of the PAL, the SwyftCard is made active in Bank 1. Accessing location \$C0B1 deactivates the SwyftCard for normal Apple //e operation.

The INH' line is driven by a tri-state driver, so if another card in the Apple //e asserts the INH' signal there will not be a bus contention. However, there will be a bus contention on the data bus if another card attempts to control the bus while the SwyftCard is active.

# SwyftCard Schematic



SWYFTCARD SCHEMATIC

## **Appendix C: List of Features**

- |   |   |
|---|---|
| 1. <b><i>Cursor</i></b>   | Blinking object on screen               |
| 1.1 <b><i>Narrow cursor</i></b>                                     | Cursor and highlight overlap            |
| 1.2 <b><i>Wide cursor</i></b>                                       | Cursor and highlight adjacent           |
| 2. <b><i>Highlight</i></b>  | Solid rectangle on screen               |
| 3. <b><i>LEAP BACKWARD</i></b>                                      | Open-Apple (held while typing pattern)  |
| 4. <b><i>LEAP FORWARD</i></b>                                       | Solid-Apple (held while typing pattern) |
| 5. <b><i>LEAP AGAIN</i></b>   | TAB while holding LEAP key              |
| 6. <b><i>Creep backward</i></b>                                     | Open-Apple (tapping)                    |
| 7. <b><i>Creep forward</i></b>                                      | Solid-Apple (tapping)                   |
| 8. <b><i>Extend highlight</i></b>                                   | Open-Apple and Solid-Apple together     |
| 9. <b><i>DELETE backward</i></b>                                    | Wide cursor and DELETE key              |
| 10. <b><i>DELETE forward</i></b>                                    | Narrow cursor and DELETE key            |
| 11. <b><i>DELETE turnaround</i></b>                                 | Occurs automatically at ends of text    |
| 12. <b><i>INSERT</i></b>  | USE FRONT A                             |
| 13. <b><i>DISK</i></b>  | USE FRONT L                             |
| 14. <b><i>COPY UP</i></b>   | USE FRONT L (and highlighted text)      |
| 15. <b><i>PRINT</i></b>   | USE FRONT N                             |
| 16. <b><i>SEND</i></b>  | USE FRONT D                             |
| 16.1 <b><i>Stop reception</i></b>                                   | USE FRONT ZS                            |
| 16.2 <b><i>Resume reception</i></b>                                 | USE FRONT ZQ                            |
| 16.3 <b><i>Send bleep to receiver</i></b>                           | USE FRONT ZG                            |
| 16.4 <b><i>Test to see if receiver<br/>        is SwyftWare</i></b> | USE FRONT ZE                            |
| 17. <b><i>CALC</i></b>  | USE FRONT G                             |

## 18. *CALC commands (require numerical input)*

<b>No.</b>	<b>Name</b>	<b>Default Value</b>	<b>Function</b>
18.1	MA%	0.	Left margin of printer
18.2	LP%	3.	Distance from bottom of page to page number
18.3	PL%	66.	Lines per page
18.4	AB%	6.	Lines in top margin
18.5	BE%	6.	Lines in bottom margin
18.6	WI%	80.	Characters per line
18.7	PF%	2.	Lowest page number to be printed
18.8	PA%	0.	Number of first page character in text
18.9	RO%		How many characters to go before text is full
18.10	SP%	1.	Line spacing
18.11	LE%	-1.	Characters transmitted at the end of every line during a SEND
18.12	SE%	5641.	A value that controls most modem parameters
18.13	BT%	5632.	The address of bottom of text
18.14	POKE	5523,0	Automatic save timer (setting equals last number times 20 seconds; maximum value: 120)
18.15	?PEEK(5630)		Bleep diagnosis
18.16	PR\$	0,0,5, 27, 77, 27, 108,8	Printer initialization (up to 29 characters)
18.17	US\$	27,45,49	Printer underline on (up to 5 characters)
18.18	UE\$	1,27,45,48	Printer underline off (up to 5 characters)

## **Keyboard**

### 19. *Cast of Characters*

- 19.1 *Lowercase letters:* abcdefghijklmnopqrstuvwxyz
- 19.2 *Uppercase letters:* ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 19.3 *Numerals:* 0123456789
- 19.4 *Punctuation:* , . ; : ' " - ? !
- 19.5 *Special symbols:* @ # \$ % ^ & \* ( ) = + [ ] { } \ | < > / ' ~

- 19.6 *Underlining*            The first underline character prints a space and turns on underlining. Underline continues until you type another underline character, two returns, or a page character. The second underline character also prints as a space.

19.7 *Format characters*

<i>Character</i>	<i>Key / Function</i>
19.7.1 Space	SPACE BAR Blank space between characters
19.7.2 Tab	TAB Inserts several spaces at a time
19.7.3 Return	RETURN Ends line, starts new line
19.7.4 Page	ESC Ends page, starts new page
19.7.5 Page break	Shows end of page (not a character)

20. *Other keys*

- 20.1 USE FRONT            Activates the commands designated on the fronts of keys.
- 20.2 CONTROL            Same as USE FRONT. Used in conjunction with Apple keys and RESET, also used to SEND control characters
- 20.3 DELETE            Erases whatever is in the highlight
- 20.4 SHIFT            Activates uppercase letters and symbols
- 20.5 CAPS LOCK            Switches all letters to uppercase, no effect on symbol keys
- 20.6 Arrow keys            Nonfunctioning in SwyftWare
21. *Emergency stop:*            **CONTROL-RESET**
22. *Emergency restart:*  
*(emergency boot)*            **CONTROL-Open-Apple-RESET**
23. *Auto-save timer*            Set timer in 20-second increments to save Text automatically; default: disabled (POKE 5523,0)

## ***Appendix D: Reassigning the Key Commands (For Experts Only)***

SwyftWare allows you to change which keys perform which commands. There is generally no need to use this feature, but software developers who wish to add features to SwyftWare may find it useful. They might design a new command and assign it to USE FRONT R, for example.

But even if you are not a software developer, you can move the PRINT command (USE FRONT N) to USE FRONT U or USE FRONT P, if it suits you. In fact, you can move it to the DELETE key or to any letter of the alphabet. You can also move a command to USE FRONT 6, -, [, ], or \. That's not all. You can assign commands to the four "cursor control" or "arrow" keys. This is because pressing a cursor control key is exactly equivalent to holding down USE FRONT and pressing certain letters.

Why this somewhat peculiar list of keys? It's actually a function of the way the Apple computer is designed, not a whim on our part. For example, in bygone days, letter keys routinely doubled as cursor control keys, and the //e and //c (among other systems) retain this wiring. That's why you can assign commands to the four arrow keys (and why the N key rather than another "home row" key such as "J" is used for the PRINT command).

### ***Talking to the Apple About Keys***

To explain to the computer what you want to do with a key, you have to refer to the key by number. The following chart shows the keys that are available for commands and their numbers. Sometimes two different keys have the same number (this is indicated in the chart). These redundancies are usually due to quirks in the design of the Apple //.

## ***Keys Available for Commands and Their Numbers***

DELETE	5536
a	5537
b	5538
c	5539
d	5540
e	5541
f	5542
g	5543
h	5544 (same as the left-arrow cursor key)
i	5545 (same as the TAB key)
j	5546 (same as the down-arrow cursor key)
k	5547 (same as the up-arrow cursor key)
l	5548
m	5549
n	5550
o	5551
p	5552
q	5553
r	5554
s	5555
t	5556
u	5557 (same as the right-arrow cursor key)
v	5558
w	5559
x	5560
y	5561
z	5562
[	5563 (same as the ESC key)
\	5564
]	5565
6	5566

### ***Talking to the Apple About Commands***

The commands are also numbered. As their names imply, some of these commands are for technical specialists. You should only play with commands you are comfortable with, and don't feel put off if there are some that don't mean anything to you.

## ***Commands and Their Numbers***

DELETE	1
PRINT	2
DISK	4
SEND	5
INSERT	7
CALC	8
CONTROL	12
TAB	27
ENTER	28

Software developers may find some of the following routines helpful.

NOOP	0	Does nothing, but returns control to SwyftWare afterward.
GET	6	This routine inserts the highlighted text of whatever disk is in Drive 1 into the Text.
MONITOR	9	Gives control to the Apple monitor program.
MAGIC	29	A location that restores the normal command-key assignments when set to zero.

Note that CONTROL-RESET gets you back from the monitor if you haven't messed around too much while in the monitor.

### ***How to Move the Command***

To move a command from one key to another you write a brief statement telling the computer what you want to do, highlight it, and use the CALC command. The statement says in effect, "I want to remove the command from this key and put it on that key."

Suppose you want to move PRINT from USE FRONT N to USE FRONT U. The first thing you need to do is write a brief statement saying you want to remove it from the N key. Look up the number for N, which is 5550, and combine it with the "do-nothing" command, NOOP. The number for the NOOP command is zero, so you write

```
POKE 5550,0:
```

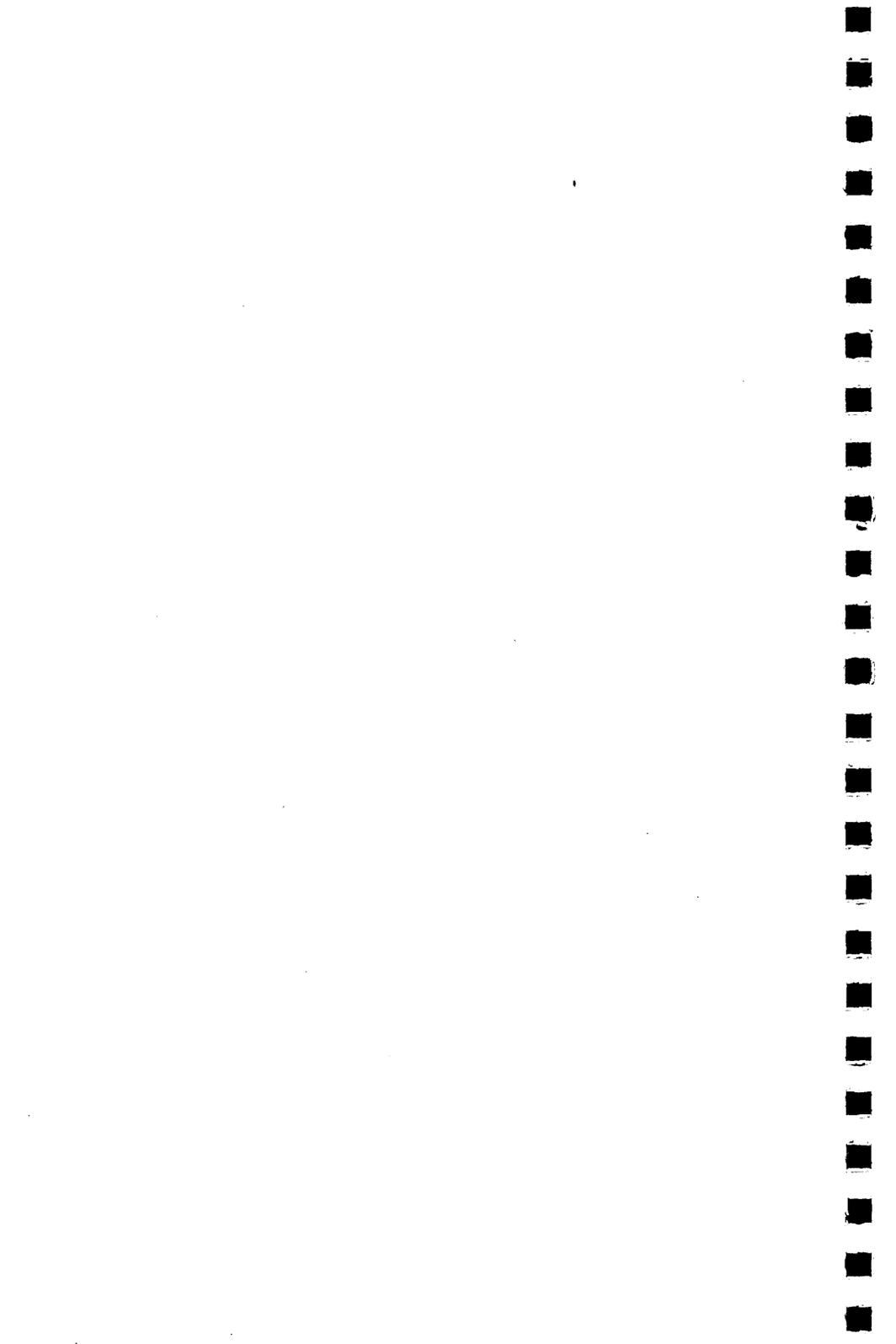
Now you have to assign the PRINT command to the U key, so look up the number for U (5557), and the number for the PRINT command (2), and then finish the statement:

POKE 5550,0:POKE 5557,2

Highlight what you have typed, and then use the CALC command. The key assignment changes instantly.

In general, if a command with the number XX is on a key with the number AAAA, and you want to assign it to the key with the number BBBB, then use the CALC command on POKE AAAA,0:POKE BBBB,XX.

***WARNING:*** *If you assign a command to G, the key that does the CALC command, you will lose the ability to do CALC at all. So remember to reassign CALC somewhere else first!*



## *Glossary*



## Glossary

### A

- AB%:** Number of blank lines at the top of each printed page; default is 6; stands for "ABove."
- accessory card:** A circuit board that, when plugged into your computer, gives the computer added capability.
- acronym:** A word formed from the first letters of the words of a descriptive phrase, e.g. "MADD" for "Mothers Against Drunk Driving."
- address:** See *machine address*.
- Apple //c:** A smaller, self-contained version of the Apple //e, with built-in peripheral interfaces rather than slots.
- Apple //e:** A later version of the venerable Apple II, a computer originally designed by Steve Wozniak.
- Applesoft BASIC:** Apple's dialect of the BASIC programming language.
- ASCII:** Acronym for American Standard Code for Information Interchange. This standard assigns a numerical code to each character. Most computers use this standard.
- autorepeat:** That ability of a key that causes it to act as if you were tapping it rapidly and repeatedly. This rapid repeat action is invoked by holding the key down for more than half a second.
- automatic highlighting:** When the PRINT, SEND, and CALC commands are invoked, the highlight will automatically extend as if both LEAP keys had been pressed at the same time and then the command had been given. The highlighted text will be the text covered in the last leap, or the text typed since the last leap. Creeping after leaping or typing will alter the boundary of the highlight.

### B

- backup, backup disk:** An extra copy made for security's sake.
- backward:** Left of or above the cursor's present location; an earlier position in the text.
- backward delete:** Erasing text to the left of the cursor a character at a time. Only a *wide* cursor will delete backward. Also see *forward delete*.
- backspace:** Moving the cursor backward by pressing the DELETE key; same as backward delete.
- BASIC:** Acronym for "Beginners All-purpose Symbolic Instruction Code." A popular computer language for personal computers. BASIC was created by Kemeney and Kurtz at Dartmouth College. Built into your Apple and accessible through your SwyftWare.
- BASIC prompt:** A sign ( ) or > ) that BASIC is awaiting input from the user.
- BASIC string:** A phrase or sequence of characters given a name in the BASIC language. Such names always end with a dollar sign (\$).
- baud:** A measure of the speed of communications, roughly equal to ten times the number of characters per second.
- baud rate:** Baud.

**BE%:** Number of blank lines at the bottom of every printed page. Default is 6. Stands for "BElow."

**BEL:** The ASCII character that, when received, causes the receiving device to emit a sound. In ancient times (say, 1965), this was a mechanical bell, and since every special ASCII character was given a three letter name, this is what they chose. If a BEL character is sent to a SwyftWare-equipped Apple, the Apple will sound a beep.

**blank:** Nothing there.

**blank disk:** A disk that contains no information.

**blank text:** Empty text.

**blankify:** Informal term meaning to make something blank.

**blinking:** Alternately appearing and disappearing. The only thing that blinks in the SwyftWare system is the cursor.

**board:** See *circuit board*.

**boilerplate:** A standard chunk of text used over and over in a document.

**bold, boldface:** A heavy-faced type; also: printing in boldface.

**boot, boot up:** Computer jargon for *start*; to load a disk and automatically run the program contained on it.

**bottom margin:** See *BE%*.

**break:** See *page break*.

**break, in communications:** A very few external systems require that you be able to put the modem into a special condition called *break*. The ability to send a break is not provided. Programmers can use BASIC to create a break condition.

**brevery:** A scanty male bathing suit.

**BT%:** Bottom of text. Used to create room for user-installed programs.

**bug:** A software malfunction.

**bulletin board:** See *computer bulletin board*.

**byte:** The amount of computer storage required to store one character.

## C

**CALC:** One of the commands (USE FRONT G). Used to do arithmetic, adjust parameters for Print and Send, and execute small BASIC programs. Only 900 bytes are allocated to BASIC. See *automatic highlighting*.

**calculation:** That art judged by a majority of the population to be second in popularity only to having one's teeth worked on; arithmetic.

**CALL 3600:** To erase a disk, type and use the CALC command on CALL 3600. After this, the disk is blank as far as SwyftWare is concerned. There is no way in the system to get back the information on the disk once this is done.

**capital:** An uppercase character; see *character*.

**CAPS LOCK:** A nuisance which, when pressed, stays down and changes all lowercase letters to uppercase. If down when pressed, the CAPS LOCK key stays up. If this key is in the down position it can cause leaping to appear to be out of order.

**card:** A circuit board.

**changed:** See *DISK*.

**character:** A building block of text. The SwiftWare system gives you 97 different characters from which to construct your text. These characters are usually classified into six groups: lowercase alphabet, uppercase alphabet, numerals, punctuation, special symbols, and formatters (for a table, see *Appendix C, List of Features*). Each character is created by pressing the appropriate key on the keyboard, sometimes in conjunction with a SHIFT key.

**character-by-character (delete):** Using the DELETE key to remove text one character at a time. This differs from using DELETE to remove a chunk of text, since text deleted a character at a time is not saved, and cannot be retrieved by the INSERT command. Deleted chunks of text can be retrieved (until the next chunk is deleted).

**checksum:** We don't usually descend into this much technicalese, but checksums are both simple and neat. Since inside the computer every character of the text is represented by a number, we can add those numbers together to form a sum (the checksum). There are many uses for checksums, for example, if at some later time we want to know if the text has been changed, all we have to do is add the character values again to get a new checksum and see if it is the same as the old one. If not, there has been a change.

**CHR\$:** A BASIC function that, given the ASCII code for a character, returns that character.

**chunk:** Two or more contiguous characters.

**circuit:** Any combination of wires and electronic parts.

**circuit board:** A sheet of material (typically fiberglass) on which wires are printed and electronic parts attached. The SwiftCard is an example of a circuit board.

**clobber:** An informal term for "destroy".

**Closed-Apple key:** The key on the right side of the SPACE BAR. The Apple logo is printed on the key in a solid color (no outline). See *LEAP FORWARD*.

**code:** A numerical designation.

**column:** Vertically aligned characters, words, or numbers; a vertical line on which characters may be aligned; see *TAB*.

**command:** An action initiated by the use of a *command key*.

**command key:** A key that will give a command to the computer if it is tapped while the USE FRONT key is held down; see *hold and tap*.

**commercial data services:** Large computers set up so that you can, for a fee, phone in and retrieve a wide variety of information, including stock quotes, news, legal references, business directories, airplane schedules, and much more. You use a data service when you send and receive Telex messages with SwiftWare.

**communications:** Transferring information between your Apple and another computer.

**computer:** A device which runs programs.

**computer bulletin board:** a computer system set up so that you can post messages to other people and see the messages they have left; an electronic kiosk.

**CONTROL:** A key on the middle left side of the keyboard, labeled "USE FRONT" in the SwyftWare system. When certain letter keys are tapped while this key is held down, the specified commands are invoked (for a list, see *Part 7, "Other Features of the Keyboard"*). Also used in conjunction with the RESET key to unfreeze the cursor (*Part 22, "Frozen Cursor"*), stop a PRINT, SEND, BASIC program, or repeating "?SYNTAX ERROR" message (see *Part 21, "To Stop a BASIC Program"*). Used in conjunction with Open-Apple and the RESET key to load a disk, overriding the DISK command (*Part 22, "Overriding the DISK Command"*).

**control character:** Special characters not used by SwyftWare, but required by some data services that can be accessed through the SEND command. Generated in the following manner: Press and hold the CONTROL key, and, while holding it down, tap the Z key and then the desired control character. The control character will be sent to whatever interface is in Slot 2. Control characters can be sent to a printer via the printer initialization technique (*Part 19, "Control of Printing Style"*).

**controller, also disk controller:** The circuit board that is the interface between your Apple and its disk drive.

**copy:** Duplicate (noun or verb). You can duplicate a chunk of text by highlighting it, deleting it, and then using the INSERT command two or more times. You can duplicate a disk by using the DISK command twice in a row, the second time on a blank or backup disk.

**copy up:** To use the DISK command to insert a highlighted chunk of text from one Text into a second Text.

**copyright:** In this case, our right to not allow you to copy our software, hardware or manuals; thank you.

**COS:** A mathematical function.

**CR:** Abbreviation for "return character" (from Carriage Return); used specifically where the ASCII code 13 is meant.

**creep:** Moving the cursor a character at a time by tapping on a single LEAP key.

**crunch:** A colloquial term for clobber.

**cursor:** The blinking object that shows where the next character you type will appear; see *leap* for cursor motion information; also see *narrow* and *wide*.

## D

**data:** Information.

**data services:** See *commercial data services*.

**debug:** To remove the bugs from; to make a program or system correct.

**default, default setting:** SwyftWare's standard setting for various parameters.

**delete:** To cause one or more characters to be removed from the text or from a pattern; erase.

**DELETE:** The key marked "DELETE". Erases the character or characters inside the highlight when pressed. When pressed while a LEAP key is being held down, removes the last character from the pattern. (See *backward delete* and *forward delete*.)

**different:** The term for a disk in the drive that is not the same disk as the one that provided the text on the screen; see *DISK*.

**disk:** A square plastic envelope that contains a round sheet of plastic coated with magnetizable particles (the same sort of stuff as recording tape) on which computer data is stored.

**DISK:** One of the commands (USE FRONT L). The DISK command loads, saves, or boots a disk, depending on which action is appropriate. If it can do none of these, it beeps. Use DISK before you remove a disk, and after you place one in the drive. When you use the DISK command, SwyftWare looks at the text, and takes the appropriate action based on whether

1. the text has been saved on disk, in which case it is called *unchanged* and it is OK to load a new text;
2. there is no text at all, in which case it is called *empty* and it is also OK to load a new text;
3. some editing has been done, in which case it is called *changed* and must be saved.

Before a load or save, SwyftWare also checks the disk to see if is appropriate for the operation. For example, it won't save on the wrong disk. There are four kinds of disks:

1. the *same* disk you are currently working from,
2. a *different* disk made by SwyftWare,
3. a *blank* disk which is always saved to,
4. or a *non-SwyftWare* disk which is booted if the text is unchanged.

Another set of conditions checked is whether there is no disk at all in the drive, the drive door is open, or the disk is *write-protected*.

**disk controller:** See *controller*.

**disk drive:** The device that stores and retrieves information from a disk.

**diskette:** Disk.

**display:** The physical device on which text is made visible.

**document:** A part of the text that you consider as being a single entity. SwyftWare itself has no concept of "document" as a special category of text.

**double-space:** To type or print on every other line.

**drive:** Disk drive.

**duplex:** Communication over a phone line may permit both people (or systems) to speak at the same time — this is full-duplex. If only one may speak at a time (as with CB radios) the system is said to be half-duplex. SwyftWare allows full-duplex communications.

## **E**

**editing:** The process of changing your text.

**editor:** A word processor.

**empty:** When SwyftWare has no text on-screen the text is termed empty; see *DISK*.

**empty text:** The state where there are no characters in the text, that is, on-screen. You can tell when the text is empty because no characters appear on the screen other than page characters 0 and 1, which are right next to each other. When deleted characters are in memory, the *text* may be empty while the *Text* is not. The test for this is to use the INSERT command and see if text appears on-screen. If the *Text* is entirely empty, using the INSERT command results in a bleep (nothing to insert).

**end of line:** See *return*.

**enter:** To give information to the computer.

**equal signs (a row of):** Indicates a page character, which signifies the end of a page.

**erase:** To delete.

**erase a disk:** See *CALL 3600*.

**ESC:** The key in the upper left-hand corner of the keyboard. It is used to put a page character into the text. To get an ASCII escape character, as used in printer setup, use CHR\$(27). To send an ESC, hold CONTROL, and, while you hold it, tap the Z key and then the ESC key.

**exponentiation:** Raising a number to a power.

**expression:** A meaningful combination of symbols.

**extend:** To enlarge the highlight so that it contains more than one character.

**extended highlight:** A highlight that contains more than one character.

**extended 80-column card:** An 80-column card (see *Numerals and Symbols* section, last page of *Glossary*) with extra memory; not needed to operate the SwyftWare system.

## F

**fanfold:** Paper made from connected sheets folded in alternate directions. Most printer paper takes this form.

**fingers:** The (usually) gold-plated strips on a circuit card which allow it to connect electrically with the computer.

**flashing:** Blinking on and off. The other definition is not suitable for a family computer.

**font:** A particular graphic design of an alphabet and other characters and symbols.

**form feed:** A signal from a computer to a printer that advances the paper so that printing can proceed on the next page.

**format:** The design features of something printed.

**format characters:** Characters that control the appearance of text on-screen and in the printout. SwyftWare has three such characters: space, return, and page.

**formatting disks:** The act of making disks usable on a system. This is never necessary with SwyftWare.

**form feed:** A signal that, when sent to most printers, tells them to move the paper so that printing will resume on the next physical page.

**forward:** Right of or below the cursor's present location; a later position in the text.

**forward delete:** Deleting text to the right of the cursor. The cursor must be narrow for a forward delete. While deleting, the cursor stands still. The characters to the right march toward the cursor, appearing to be "vacuumed" up.

**full duplex:** See *duplex*.

## G

**gadgetry:** See *paraphernalia*.

**global:** Everywhere. A global search and replace means to find a term wherever it occurs in the text, and to replace every occurrence.

**graphics, graphics output:** Pictures drawn with a computer. Using BASIC you can draw pictures with SwyftWare.

## H

**half duplex:** See *duplex*.

**hang:** A bat's sleeping posture. In the jargon of the realm, the computer "hangs" when it stops responding to anything you can do from the keyboard.

**hard copy:** A printout in paper and ink of the things you write and store electronically in the computer.

**hard disk:** A device that can store and retrieve more information more quickly than a floppy disk.

**hardware:** The physical parts of a computer system.

**highlight:** (v) To mark a chunk of text so that something can be done to it all at once. Some of the things you can do to a highlighted chunk of text include deleting, printing, calculating, moving, and sending. (n) The unblinking rectangular object that accompanies the cursor. Indicates which character or characters are highlighted. Anything in the highlight will be erased when the DELETE key is pressed.

**high resolution:** "I promise to climb Mount Everest next year." In computerese, however, a relative term that describes a system that can draw more dots (i.e. finer or more numerous characters) than a system that has lower resolution.

**hold and tap:** A basic physical technique used extensively in SwyftWare. While one key is pressed and held down continuously, other keys are tapped. Four keys are used this way in SwyftWare: SHIFT, which enables you to type capitals and certain symbols; USE FRONT, which enables you to use the commands; and the two LEAP keys, which enable you to move the cursor.

**home row:** The row of keys on a keyboard where touch typists rest their fingers. Normally the row: ASDFGHJKL ; '

**homily:** An inspirational catchphrase.

## I

**Incantation:** A written spell or charm that will produce certain effects inside SwyftWare.

**Indent:** To set in from the margin; see *TAB*.

**Indentation:** How far the first line of a paragraph is set in from the margin.

**Information retrieval:** Giving a system a clue as to what you are looking for and having the system try to find it. Information retrieval also includes the process of entering information into a system in such a manner that the information can be retrieved.

**Information, storage of:** Your text may be stored in the computer, on a disk, or both.

**Initialize, initialization:** A step you have to take with some devices before you can operate them. This applies especially to printers and modems.

**Initialization sequence:** The list of things you have to do to get something started. Printers and modems must sometimes be initialized.

**INSERT:** One of the commands (USE FRONT A). If a chunk of text has been deleted by a single tap of the DELETE key, INSERT will recall it to the screen. INSERT is used to undo accidental deletions, copy text, and move text from place to place. The deleted material is available for recall until another chunk is deleted.

**Instance:** An example of something.

**Interface:** The software and circuitry required so that two devices (or one device and a human) can communicate; somewhat trendy if used to mean talking to another person.

**interface card:** See *interface* and *card*.

**Interrupt:** A signal that alerts the computer to something happening.

**Invoke a command:** To use a command.

**Italic:** A typographic style that features slanting rather than vertical letters.

## K

**K:** Short for kilobyte. One K equals 1024 bytes.

**key:** One of the pushbuttons on the keyboard.

**keyboard:** A device with many push buttons, designed for tiring your fingers. You use it to control and communicate with your computer.

**keyword:** A word or phrase that sticks out in your mind as being associated with some document or place in a document to which you'd like to move the cursor.

## L

**land on:** To occupy the same position as.

**leap:** The term that describes SwyftWare's cursor motion. To leap to the target character, you hold down a LEAP key, then type a

pattern consisting of the target character followed by the characters that come after the target character in the text. The cursor will leap to the nearest instance of that pattern. If the pattern cannot be found, the cursor will remain where it is. The leap mechanism is patent pending.

**LEAP AGAIN:** Alternate function of the TAB key, available when a LEAP key is held down; causes the cursor to leap to the next instance of whatever pattern you have typed without you having to retype the pattern.

**LEAP AGAIN key:** The key marked "TAB" on the left edge of the keyboard, when used in conjunction with a LEAP key; autorepeats when held down.

**LEAP keys:** The cursor control keys, Open-Apple and Solid Apple, which are located immediately to the left and right of the SPACE BAR. Pressing and holding a LEAP key readies the cursor to leap to a pattern of characters anywhere in the text. Tapping a LEAP key causes the cursor to *creep*, that is, move one character at a time. Pressing both LEAP keys at once extends the highlight. Most typists prefer to operate the LEAP keys with their thumbs.

**LEAP BACKWARD:** The Open-Apple key on the left side of the SPACE BAR. Pressing and holding this key readies the cursor for a backward leap through the text. See *backward*.

**LEAP FORWARD:** The Solid-Apple key on the right side of the SPACE BAR. Pressing and holding this key readies the cursor for a forward leap through the text. See *forward*.

**leap pattern:** See *pattern*.

**LE%:** Sets the ASCII code of the character to be sent out at the end of every line when using the SEND command. If a return is desired, set LE% to 29. Default is -1, which means nothing is sent out. Stands for "Line End."

**left margin:** See *margin*, *PR\$*, and *MA%*.

**LF:** Abbreviation for "line feed," used specifically where the ASCII code 10 is meant.

**line:** A horizontal row of characters, or the space reserved on the screen or paper for a row of characters.

**line feed:** Moves the printer up one line.

**line width:** See *W1%*.

**LIST:** A BASIC command that displays a program you have written.

**load:** To bring the information on a disk into the computer. No information is removed from the disk during a load; rather, the disk and feeds it into the computer. See *save*.

**location:** A place. In computer jargon, usually a place in memory.

**LOG:** A mathematical function.

**lost:** Leaping to a different part of the text than you intended. If you have not let go of the LEAP key, you can get unlost just by tapping a few characters at random.

**lowercase:** Letters that are not capitals. Lowercase letters in a leap pattern match both uppercase and lowercase letters in the text. See *character*.

**LP%:** The number of lines the page number will appear above the

bottom of the page. Default is 3. Stands for "Latitude of Page number."

## **M**

**machine address:** A number (much like a house number) that tells where a particular piece of information is stored in a computer.

**manual:** The second most important part of any computer system (you're the most important part.)

**margin:** The space between the top or sides of a piece of paper (or the screen) and the borders of the printed material; see *PR%*, *AB%*, *BE%*, *MA%*, *WI%*, and *PL%*.

**MA%:** The number of spaces between the left edge of your text and the leftmost printing position of your printer. Applies to text on the printer only; does not affect the appearance of text on the display. Changing MA% changes the right margin; some adjustment of WI% (see also) may be necessary as a result. Default is 0. Stands for "MArgin."

**memory:** The part of the computer that is capable of storing information.

**mistake:** Something you do which proves you are a human being.

**modem:** A costly little box or circuit card that allows your computer to talk to other computers. The name is an acronym for MOdulator/DEModulator, so-called because it takes the computer signals and turns them into sounds that can be sent over telephone lines (modulation), and turns the sounds from other computers into signals your computer can understand (demodulation.)

**modem card:** An interface card for a modem, or a complete modem; installed in Slot 2 of the Apple //e.

**monitor:** The device that has the screen that you stare at for hours on end. It is usually sort of a TV.

**move text:** To delete text from one place and INSERT it at another.

**munch:** To devour.

## **N**

**narrow:** A state of the cursor when the blinking cursor and the solid highlight are both on the same character. The cursor becomes narrow after a leap or creep. A narrow cursor forward deletes. See *wide*.

**NEW:** A BASIC command.

**non-SwyftWare disk:** A disk that is not programmed in the SwyftWare system (see *DISK*).

**no disk:** An empty disk drive (see *DISK*).

## **O**

**on-line:** Available via a telephone call or from the computer immediately.

**on-screen:** Visible on the monitor screen.

**Open-Apple key:** The key to the left of the SPACE BAR, used in conjunction with character keys to leap the cursor backward in the text; see *LEAP BACKWARD*.

**open space:** Scarce commodity; places on the screen where there are no characters, not even invisible characters such as spaces or returns.

**Outline-Apple key:** The Open-Apple key.

## **P**

**PA%:** The page number of the first page character in a Text. All numbers are determined in relation to this page number. Default is 0. Stands for "Page."

**PAGE:** The key that produces page characters. This key is marked "ESC" and should be labeled "PAGE" by the user of SwyftWare. See *page character*.

**page break:** A dashed line that symbolizes the end of a page. The number in the middle of the line is the page number of the text above it. SwyftWare puts in a page break whenever you create enough text to fill a printed page. A page break is not a character, hence you cannot delete it, leap to it, or produce it with a key. See *page character*.

**page character:** A character that forces the end of a page. The page character symbol is a double-dashed line with the page number in the middle. Since it is a bona fide character — just like a letter, numeral, or punctuation mark — you can type it, delete it, or leap to it. See *PAGE*, and *page break*.

**page number:** The number that appears in the middle of page characters, and which will print at the bottom center of the paper page; see *LP%*, *PA%*, *PF%*.

**pagination:** The arrangement of page boundaries in the text.

**paragraph:** In SwyftWare, a chunk of text bordered by blank lines or enclosed by page characters.

**parameters:** A number or value that is needed by some part of the system, such as line width, number of lines per page, margins, baud rate, or page numbering.

**paraphernalia:** See *gadgets*.

**parity:** A virtue that goes along with faith and hope. (Sorry.) Parity is a property of signals sent between computer systems. It can be even or odd, or there can be no parity. What it is is not nearly as important as making sure your system has the same parity as the system with which it is communicating. Most bulletin boards and data services have no parity.

**pathname:** A complete ProDOS file name including the volume name, subdirectory names (if any), and the file name.

**pattern:** One or more characters you type while holding down a LEAP key. The cursor leaps to the nearest instance of the pattern, landing on the first character (the *target character*). A pattern can be up to 31 characters.

**perforations:** The place fanfold paper is supposed to tear.

**peripherals:** Accessories for a computer.

- PF%:** The lowest page number that you want printed. Default is 2, meaning that. Page numbers lower than 2 will not print. Stands for "Page Floor."
- PL%:** The number of lines per page. Default is 66.
- power:** A corrupting influence.
- power supply:** That portion of an electronic device that converts whatever form of electricity is available in your locale into the form of electricity required by the circuitry of your computer.
- power up:** Computer jargon for "turn on."
- PR\$:** See *printer initialization*; stands for "PReinter."
- PR#6:** A command used to "boot" an Apple. May you not have to use it.
- preboot disk:** An abomination required by some non-SwyftWare products. Loads a program that modifies the computer so that you can run another program.
- press:** To push down a key on the keyboard; see *tap*.
- print:** To translate text on screen into ink on paper.
- PRINT:** One of the commands (USE FRONT N); prints the text that was covered in the last leap, or the text that was typed since the last leap, or whatever text is highlighted; see *automatic highlighting*.
- print buffer:** Some printers or interfaces have the ability to store a lot of text very quickly so that you can go back to using your computer while the printer completes its task. The place that the text is stored in the printer or interface is called the print buffer.
- printer:** A device that reproduces your text on paper (provided it feels like working that day).
- printer card:** An interface card that allows the Apple to talk to a printer; placed in Slot 1.
- printer initialization:** A task that must be done to make some printers operate properly. SwyftWare uses variables called PR\$, US\$, and UE\$ for initialization. Once you put the required information into them, SwyftWare automatically sends them to the printer each time you use the PRINT command. The printer initialization is saved along with your disk when you use the DISK command. By using the CHR\$ function in BASIC, any ASCII code (including ESC and control characters) can be put into PR\$ and thus sent to the printer. Default is Epson FX-80 or compatibles. See *variable, CHR\$*.
- print head:** The part of the printer that actually does the printing.
- print out, printout:** The act of printing; the stuff you get from a printer.
- program:** (v) to write instructions for the computer; (n) a set of instructions that tells a computer how to accomplish a task.
- programmer:** A person who either no longer needs other humans for solace or companionship, or who doesn't realize this until they have all gone to sleep.
- protocol:** The art of being obsequious. An advance agreement between two parties to accomplish a certain task. For example, the protocol for communicating between SwyftWare and a Hayes modem is that all commands begin with an "AT" and end with a return.

## Q

**quandary:** A puzzling situation.

## R

**random:** The technique used to write this glossary.

**read:** To access information, as from a disk.

**retrieve:** To find something tucked away. The leap technique is an excellent retrieval mechanism. See *information retrieval*.

**RETURN:** The RETURN key; types return characters.

**return:** One of the characters you can type. On SwyftWare, the return character puts the cursor on the next line and moves it to the left edge. After a return, the next character you type will display and print on the next line at the left border of the text.

**right margin:** See *WI%*.

**RO%:** The number of characters you can type before your Text is full. "RO" stands for "ROom."

**routine:** A set of instructions for doing a small task.

**RUN:** A BASIC command.

## S

**scrolling:** Moving all the text on the screen up or down. When this is done repeatedly and rapidly, the whole screenful of text appears to be moving up or down as if on a winding or unwinding scroll of paper.

**SEND:** One of the commands (USE FRONT D). Transmits text through an Apple Super Serial Card in Slot 2, and out the serial port. The text sent will be the text covered in the last leap, typed since the last leap or received text, or highlighted at the time the command is used. See *automatic highlighting*.

**serial port:** Either the act of drinking one glass of wine after another, or a destination for ships laden with grain. Also, an electronic device which allows your computer to communicate with another device that also has a serial port. Serial ports are usually found on modems and sometimes on printers.

**set:** To give a parameter a value. This is done by using the CALC command on a formula that includes a variable, an equal sign, and the numerical setting. For example, use the CALC command on  $WI\%=60$  to set the line width to 60 characters (*WI%* is the variable.)

**setup:** Initialization.

**SHIFT:** One of the two keys marked with the word "SHIFT". Typing while holding either one of them gives you uppercase letters and the upper symbol on those keys which have two symbols.

**Sider:** A brand of hard disk.

**SIN:** A mathematical function.

**single-space:** Lines of text (or print) that follow each other with no blank lines between.

**slot:** One of the seven places you can plug accessory cards into an Apple //e.

**software:** Programs, i.e. instructions that tell a computer what to do.

The phrase "software programs" is redundant.

**Solid-Apple key:** See *Closed-Apple key*.

**SP%:** Line spacing in the printout. Set SP%=2 for double-spacing; set SP%=1 for single-spacing. Default is 1. Stands for "SPacing."

**space:** The only character in the English language more common than the letter "e". Put into the text by pressing the SPACE BAR. A space is a character just like any other character in the SwyftWare system.

**SPACE BAR:** One of the best scenes in the movie "Star Wars." Also the biggest key on the keyboard.

**speech synthesizer:** A device that makes text sent by a computer audible. Most can only approximate normal pronunciations. They are especially useful for the seeing impaired.

**start bits:** A communications parameter.

**stop bits:** Another communications parameter.

**store:** (v) Putting information into memory, or onto a disk.

**string:** A sequence of characters.

**string variable:** A BASIC variable that contains characters.

**stuck cursor:** Either a cursor that is flashing but will not move, or a cursor that stops flashing.

**Super Serial Card:** An Apple Computer Inc. interface card required for use with communications on the Apple //e.

**SwyftCard:** An accessory card for the Apple //e that contains SwyftWare.

**SwyftDisk:** The Apple //c or Apple //e with extended 80-column card version of SwyftWare.

**SwyftWare:** A program that allows you to do word processing, calculations, communications, information retrieval, and short BASIC programs.

**symbol:** A raucous musical instrument of the percussion family.

**SYNTAX ERROR:** A message inserted into the text by BASIC when it detects something it can't understand.

**system:** Any collection of items and ideas that can be usefully thought of as a single entity. For example, the SwyftCard system consists of a computer, its peripherals, and a circuit card that embodies certain ideas that make a computer easier to use.

## T

**TAB:** A key for typing several spaces at once. Each tap on the TAB key inserts enough spaces to move the cursor to the next column.

TAB columns are five spaces apart. See *column* and *AGAIN*.

**TAN:** A mathematical function.

**tap:** To quickly depress a key as far as it can go and immediately release it.

**target:** Target character.

**target character:** The character you want to move the cursor to; the first character in a pattern. It may be any letter, punctuation

mark, numeral, symbol, space, return, or page character, anywhere in the Text.

**tear bar:** The part of the printer that braces the paper while you tear along the perforation.

**text:** All the characters that you can see on your screen or that can be reached by a leap.

**Text:** The text taken as a whole; all the information in the computer at one time.

**thumb:** The best finger to use on the LEAP keys; may also be held down on a LEAP key to hold your place in the text, much as you might use a finger to mark your place in a book while you scan through the pages.

**top margin:** See *AB%*.

**track:** A certain portion of the information stored on a disk.

**track number:** The number assigned to a particular track.

**trademark:** A word ruined for ordinary discourse in the pursuit of commercial interests.

**tutorial:** An instructional text.

**typing:** (1) The process of striking keys on a keyboard in order to create text. (2) An undeservedly low-paying job.

**typewriter:** An ancient instrument of torture that would penalize you by ruining a whole page when you merely mistyped a few characters.

## U

**UE\$:** The codes sent to the printer to turn off underlining. Stands for "Underline End". See *printer initialization*.

**unchanged:** A state of the text recognized by the DISK command. Unchanged text has not been altered since it was last saved or loaded.

**undo:** To nullify something you have done. Deletion can be undone (with the INSERT command), highlighting text can be undone (by pressing either LEAP key), and an incorrect cursor leap can be undone (by typing a few letters at random before you've let go of the LEAP key).

**uppercase:** Capital letters. Uppercase letters in a pattern match only uppercase letters in the text. See *character*.

**US\$:** The codes sent to the printer to turn on underlining. Stands for "Underline Start". See *printer initialization*.

## V

**vacuum:** The way delete seems to work when the cursor is narrow. See *forward delete*.

**variable:** A short sequence of characters, such as *W1%*, *I*, or *PR\$*, that can represent a number or a string in BASIC. See *the Apple BASIC manual*.

**video:** The signal going to the monitor.

**volume:** ProDOS name for a disk.

## W

**whir:** A polite word for the dreadful sound the disk drive makes when it starts up.

**WI%:** Characters per line. Set WI% to the maximum number of characters you want in each line, both on the screen and on the printer. The minimum is 16, the maximum (and default value) is 80. A change in MA% will affect the right margin and may require an adjustment in WI%. "WI" stands for "WIDTH".

**wide:** A state of the cursor. The highlight and cursor are split apart (not on the same character), and the highlight contains only one character. See *narrow*, *backward delete*, and *extended highlight*.

**width:** The maximum number of characters permitted in a line. See WI%.

**window:** The portion of the text that you can see in the display.

**word:** (1) In SwyftWare, a consecutive sequence of non-blank characters and up to two of the spaces that follow them. This is similar to the normal use of "word" except that English words don't include the spaces. (2) In computer jargon, all the information at one machine address. (3) In communications, a package of information that conveys a single character.

**word processing:** Creating and editing text electronically; usually includes the means for storing and printing the text so created. All word processors make rearranging text far easier than on a typewriter.

**word wrap:** The process of automatically breaking text into lines without your having to type a return. Word wrap keeps words whole, it never breaks them in two.

**write:** To store information; information "written" onto a disk is stored electronically on that disk.

**write-protect:** To install a safeguard on a disk that prevents the disk from being altered by the computer.

**write-protect notch:** A usually rectangular notch in the right side of the disk (when the label is in normal reading position) that can be covered with a write-protect tab in order to write-protect the disk.

**write-protect tab:** A piece of adhesive-backed paper or foil which is wrapped over the edge of the disk, covering the write-protect notch. Once this is done, the text on the disk cannot be altered by the computer.

## ***Numerals and Symbols***

**8-bit word:** See *word*. Entries 2 and 3 apply.

**80-column card:** A circuit board that allows your Apple //e to show 80 characters of text per line, instead of the measly 40 you get without it.

**80-column display:** A system capable of showing 80 characters in each line.

- ? Placed in front of arithmetic expressions you want to calculate. For example, to calculate  $2 + 2$ , type ?  $2 + 2$ , and use the CALC command on this expression. Also placed in front of variables in order to ascertain their value. For example, to see how many characters you have per line, highlight and use the CALC command on ?Wl%.
- ;  
Placed after arithmetic expressions you want to calculate if you don't want a return automatically inserted afterward.



*Index*



## Index

### A

AB%, 61, 75, 119  
accelerator card, 4  
addition, 51  
advance the paper to next page,  
  see *page*  
answering the phone, 82  
Apple //c  
  equipment checklist, 5  
  using built-in disk drive, 10  
Apple //e  
  equipment checklist, 3  
  interface cards, 66, 67  
  slot assignments for Apple  
  cards, 4  
Apple parallel printer, 66  
Apple printers  
  Imagewriter, 69  
  Laserwriter, 69  
  Scribe, 69  
AppleSoft BASIC, 52, 56, 69,  
  also see *BASIC*  
Apple software, using, 47  
Apple Super Serial Card,  
  see *Super Serial Card*  
arithmetic, 51-52, 56  
arrow keys, disengaged, 20, 121  
ASCII, 74-76  
assembly language, 113  
asterisk, used for  
  multiplication, 52  
auto-answer, 82, 87-88  
autoextension of highlight,  
  see *highlight*  
autorepeat  
  def., 20

### B

backslash, 100  
backup disk, see *disk, backup*  
backward, see *leaping and delete*  
banks of switches, 66, 79  
bank switching, 115  
BASIC, 55-56  
  interpreter, 114-115  
  size of program, 55

baud rate for printers, 67  
  calculation, 83  
  for communications, 82-84  
  generator, 84

BE%, 61-62

BEL, 87

bells and whistles, 111

birthday, 101

blank disk, see *disk, blank*

bleeps

  CALC command and, 94-95  
  communications, 86-87, 95

  CONTROL G, 95

  copy up, 94

  DISK command, 41-42, 94

  empty text and, 50

  full text, 44, 48, 93

  INSERT command, 93-94

  SEND command, 95

  table of, 96-97

  while typing, 93

Bleep Table, 96-97

board, see *printed circuit board*  
boldface, 77

boot procedure, 91

boot the system, see *restarting*

bottom margin, 61

Brother printers

  HR-10, HR-15XL, HR-25,

  HR-35, 70

  M-1009, 69

  Twinriter 5 (DP dot matrix  
  mode), 69

  Twinriter 5 (WP daisy wheel  
  mode), 70

  2024, 69

BT%, 115

bulletin boards, electronic,  
  79, 86

bus contention, 116

### C

CALC

  autoextending, 33, 51

  command, 51-52

  command number, 122-123

  key, 9

  reassigning warning, 124

  setting formatting values  
  with, 60

calculations, 51-52  
calendar, 100-101  
CALL 3600, 46-47, 55  
capital letters  
  in leap patterns, 28  
  typing of, 16  
CAPS LOCK key, 19-20  
  interfering with LEAP, 28  
card, see *printed circuit board*, 3, 7-8  
carriage return,  
  see *return*, 66, 74  
carrier sheet, 9  
CAUTION  
  deleting entire text to  
  force load, 43  
  turning off computer  
  to start over, 10  
  using CONTROL-SOLID-  
  APPLE-RESET, 43  
  using ProDOS converter,  
  104, 107  
characters  
  available for commands,  
  121-122  
  cast of, 17  
  format, 17  
  what they are, 15  
checksum, 97  
CHR\$ used in printer  
  initialization, 74-75  
chunk of text  
  def., 32  
  how to highlight, 32  
  lifetime when deleted, 37  
  typing and highlighting, 32-33  
C-Itoh (Apple parallel printer), 70  
clobber, 91, 113  
Closed-Apple key, see *Solid-Apple*  
code, 74-77  
commands, assigning to new keys,  
  121-124  
communications, see *SEND*  
communications, garbled, 85  
control codes to printer,  
  see *printer control codes*  
  to data service or bulletin  
  board, 86  
  to modem, 86

CONTROL  
  command number, 123  
  CONTROL C, 56  
  CONTROL E, 87  
  CONTROL G, 87  
  key, 9, 20  
controller; for disk,  
  see *disk controller*  
CONTROL-OPEN-APPLE-  
  RESET, 91  
CONTROL-RESET, 56, 91  
conversation, electronic,  
  79-81, 87-88  
convert, see *ProDOS*  
  *Converter*  
copy up, 44-45  
  accidental, 45  
  illus., 45  
  use in splitting Text, 50  
correspondence quality, 77  
COS, 52  
CR, 67, also see *return*  
creeping  
  forward and backward, 25  
  general, 24  
  to adjust highlight, 33  
cursor  
  blinking part, 15  
  control keys, lack of  
  need for, 111  
  control key commands, 121  
  creeping, 24-25  
  definition of, 15  
  design, 112  
  effect of width on delete, 35  
  leaping, 24-25  
  moving, see *leaping*  
  and *creeping*  
  narrow, 35  
  unblinking part, 91  
  wide, 35

## D

data bits, 83-84  
  services, 79, 85-86  
Data Products 8010, 8050, 70  
debugging the printer, 77  
default  
  default values, 119  
  also see *specific topic*

meaning of, 65  
delete  
buffer and mode of  
operation, 114  
chunks, 36  
direction of, 35  
(also see *cursor, wide;*  
*cursor, narrow*)  
entire text, 43  
how much you can, 36  
relation to highlight, 15  
single characters, 35  
turnaround, 36  
undoing, 36  
unlimited amount, 36

DELETE  
command number, 123  
correcting pattern errors  
and, 27  
key, 19

desert  
writing in sand on  
windblown, 39

Diablo 630, 70

dialing, 82

disk  
backup, 46, 56, 114  
blank, 3, 5, 40  
controller, 3  
how to handle, 39  
how to make blank, 46  
protecting, 41-42  
relation to text, 39  
switching to new text, 41

disk drive, 3, 10  
how many you need, 39  
how to find drive  
number, 39

diskette, see *disk*

DISK  
automatic timer and, 54  
beeps and, 42  
command, 39, 50, 112  
command number, 123  
copy up and, 44-45  
importance of using  
frequently, 40  
key, 9

division, 52

document, page character  
used to separate, 21-22

DOS (Disk Operating System),  
lack of need of, 112, 114  
double-spacing, 64-65  
draft quality, 77  
drain (static electricity), 7  
drive, see *disk drive*  
DTC, 70

## E

emergency restart, 91-92  
emergency stop  
from BASIC, 56-89  
when sending, 87, 90  
while printing, 59, 89  
while receiving SEND,  
87, 89-90

empty text  
def., 13  
from SwyftCard, 13  
from SwyftDisk, 14  
illus., 13

ENTER, 123

Epson printers  
FX-80, FX-80+, FX-85,  
FX-100, FX-185, RX-100, 69  
LQ-1500, SQ-1500, 70

equipment checklist  
for Apple //c, 5  
for Apple //e, 3  
for ProDOS conversion, 102

erasing a disk, see *CALL 3600*  
(erasing text, see *delete*)  
error in a BASIC program, 89, 91  
handling in ProDOS  
converter, 110

ESC key, 9, 75-76  
also see *PAGE*

exponentiation, 52

## F

features, list of, 118-120  
file, 102-103  
first page of a document  
unnumbered, 62  
font, 76-77  
force loading, 43-44  
format characters,  
see *characters, format*

format disks, lack of  
  need to, 111, 114  
formatting printing,  
  see *print formatting*  
form feed, 58  
FORTH, 113, 115  
forward, see *leaping and delete*  
forward and backward  
  illus., 24  
  where they are, 24  
frozen cursor, 91  
Fujitsu Dot Matrix 24, 70  
full text, what to do, 48-50

## G

gap, 113  
garble, 85  
GET, 123  
getting started (without  
  tutorial), 13  
Glossary, 125  
Golden Disk Rule, 39  
Gothic, 75  
graphics, 52  
ground connection, 7

## H

habit formation, 112  
hard disk, 47  
hardware theory of  
  operation, 115-116  
Hayes, 80  
Hewlett Packard printers  
  LaserJet, 71  
  ThinkJet, 69  
highlight  
  adjust by creeping, 33  
  automatic extension of, 33  
  autoextending in CALC, 51  
  boundaries of extended, 32  
  definition of, 15  
  direction of leap and, 32  
  how to extend, 32-34  
  rehighlighting, 33  
  undo, 33  
hold, definition of, 16  
hoops, hula, 99  
hula hoops, see *hoops, hula*

## I

IBM Graphics Printer, 69  
IDS, see *Integral Data Systems*  
Imagewriter, 69  
information retrieval,  
  98-99  
INH', 116  
initialization, see *printer initialization*  
  avoid redoing, 77-78  
  save time typing, 73  
insert  
  by typing, 17, 37  
  versus replace, 17  
INSERT  
  chunks, 37  
  command, 37-38  
  command number, 123  
  key, 9  
  lifetime of deleted  
    chunk, 37  
  used in search and  
    replace, 38  
  used to make copies, 37  
insertion of text, 15, 17, 37  
installation  
  of interface cards, 4  
  of SwyftCard, 7-8  
Integral Data Systems  
  480, 71  
integrated circuit, 115  
interface, 4  
interface cards, 66  
interface theory of  
  operation, 111-113  
interrupting communications, 87  
inventories, 99  
italics, 75, 77

## J

Juki 6100, 6300, 71

## K

keyboard features, 15-20  
keyboard labels, see *labels*  
key commands, reassigning,  
  121-124

## L

- labels for keyboard, 3
  - where to place on keyboard, 9
- Laserwriter, 69
- LE%, 81-82
- LEAP AGAIN
  - command, 27
  - key, 9
- LEAP
  - command, 24-31, 112
  - key, 9, 24
  - undoing highlight, 33
- leaping
  - always goes to first character of pattern, 27
  - as information retrieval, 98-101
  - circularity of, 27
  - correcting pattern errors, 27
  - directionality of, 24
  - document-by-document, 30
  - effect of capital letters on, 28
  - forward and backward, 24
  - general, 24
  - how to, 25
  - letting go and, 26, 27
  - most common mistake in, 26
  - paragraph-by-paragraph, 29
  - sentence-by-sentence, 29
  - speed of, 24
  - to define chunks, 30
  - to open space, 30
  - undoing, 26-27
  - use of autorepeat, 27
  - word-by-word, 29
- left margin of printer, 60-61
- letting go, importance of, 25
- LF, see *line feed*
- line feed, needed by printer, 67
- line breaks, erroneous, 68
- LIST, 52, 55
- LM311, 115
- LOG, 52
- logic board, see *printed circuit board*
- LP%, 64

## M

- MA%, 60-61
  - relation to WI%, 61
- MAGIC, 123
- mail lists, 100
- mail merge, 100
- Mannesman Talley 160L, 71
- margins, 60-62
- menu, 85, 102, 111
- message center, 87-88
- message to printer,
  - see *printer message*
- message from source
  - outside computer, 85-86
- messages, stored, 53-54
- modelessness, 112
- modem, 80
- modem interface, 79
- modes, 112
- MONITOR, 123
- monitor, 3, 5
- monotony, 112
- mouse, lack of need for, 111
- moving
  - chunks of text, see *insert cursor*, see *leaping* and *creeping*
  - text from one disk to another, see *copy up*
- multiplication, 52

## N

- NEC Spinwriter, 71
- NOOP, 123
- noun-verb principle, 113
- numerals, 17

## O

- Octave Systems (DTC)
  - Laser Printer, 71
- Okidata 192, 193 printers, 72
- on-screen instruction, 10-11
- Open-Apple key, 9, 24
- open-circle label, 9
- open space
  - def., 30
  - illus., 30
  - leaping to, 30

operating system, lack of  
  need for, 111  
options, in ProDOS  
  conversion, 102-110  
overriding the DISK  
  command, 91

## P

PA%, 63  
page  
  advance paper to next, 21  
  changing numbering of, 63  
  maximum number of, 23  
  numbering, 23, 59, 62-64  
  number of lines per, 21, 62  
  number, position of, 64  
page break  
  different from page  
    character, 22-23  
  graphic representation  
    of, 21  
  inserting text above,  
    22-23  
  maximum number, 23  
  underlining and, 23  
page character  
  definition of, 21  
  different from page  
    break, 22  
  first and last not  
    erasable, 22  
  illus., 21  
  inserting text above, 22  
  turning off underlining, 19  
  used to begin document,  
    21-22  
  used to end page, 21  
  used to ensure top of  
    page, 21-22  
  used to improve  
    appearance, 22  
  used to shorten a  
    page, 22  
PAGE key, 9, 21  
pagination, 21-23  
PAL, 115  
Panasonic  
  KX-P3151, 72  
  P-1092, 69  
paradigm, 111  
parity, 84  
pathname, 104  
patterns  
  correcting, 27  
  definition of, 25  
  maximum length of, 25  
  nonexistent, 26  
  retention, 28  
  reuse of, 28  
PEEK, incorrect use of, 55  
people who made SwyftWare,  
  see *Credits*  
perforations in paper, 21  
PF%, 62  
phone, use of, 82-83,  
  87-88  
phrase storage, 53-54  
PL%, 62  
poetry, see *verse*  
POKE, incorrect use of, 55  
power  
  failure, 40-41  
  light, 7  
  power-on reset, 116  
  switch, 7  
  using DISK before  
    turning off, 39-40  
  warning about  
    turning off, 10  
PR\$, 67-68  
PR#6, 47-48  
PRINT  
  command, 57-59  
  command number, 123  
  key, 9  
  with autoextend,  
    33-34, 57  
print  
  a character, 57  
  a document, 58  
  an entire Text, 58-59  
  one page at a time, 58  
  what is highlighted, 57  
print formatting  
  bottom margin, 61-62  
  control of style, 75-76  
  font selection, 76-77  
  lines per page, 64  
  line spacing, 64  
  line width, 60  
  local font selection, 76-77

- left margin, 60-61
- lowest page number, 62
- page numbering, 62-63
- position of page number, 64
- questioning values, 65
- resetting values, 65
- style, 75-76
- top margin, 61-62
- underlining control, 76-77

printer

- brands, 69-73
- cable, 3, 5
- control codes, 73-77
- control of in general, 74-75
- debugging, 68, 77
- defaults, 69, 119
- erroneous line breaks and, 68
- form feeds and, 58
- interface, 57, 66
- initialization, 67
- initialization transferred to new Text, 77-78
- message for setup, 73, 75
- not requiring setup, 69
- questioning values, 65
- requiring setup, 69-73
- resuming, 59
- setup, 66, 73-74
- settings for specific printers, 69-73
- stopping, 59
- switch settings, 66

printout, 57-58

problems with current systems, 111

ProDOS converter, 102-110, 114

- AppleWorks and, 110
- DOS 3.3 and, 110
- equipment, 102
- errors and, 110
- how to boot, 102
- menu, 103
- menu illus., 103
- one disk drive, ProDOS to SwyftWare, 106
- one disk drive, SwyftWare to ProDOS, 108

- oversized texts and, 110
- two disk drives, ProDOS to SwyftWare, 103
- two disk drives, SwyftWare to ProDOS, 105

programming, 52

PROM, 115

## Q

Qantex 7030 printer, 72

question mark

- use in calculation, 51
- used to interrogate values, 65

Qume Sprint 5, 9, 11, 11 Plus printers, 72

## R

RAM, 114-115

reassigning key commands, *see key commands*

receiving information via modem, 86-87

registers of 6551, 85

rehighlighting, 33

required equipment

- Apple //c, 5
- Apple //e, 3

RES signal, 116

RESET, *see CONTROL-RESET*

resetting format values, 65

restarting the system, 91-92

return

- at end of every line when SENDING, 81, also *see LE%* illus., 18
- needed by printer, 74
- not needed at ends of most lines, 16
- two will stop underlining, 19
- use of, 17-18

ringing phone, 82

RO%, 53

ROM, 115

room

- left for typing, 53
- what to do when you run out, 48-50

RUN, 52

## S

save

- automatic timer, 54
- importance of, 10, 40-41
- first time, 40

schematic diagram, 117  
description, 115-116

scrolling, 29

SE%, 83-85

search and replace, 38

SEND

- a bleep, 87
- automatic highlight,  
33-34, 80
- command, 79-88
- command number, 123
- control codes, 86
- key, 9
- return at every line  
end, 81
- stopping and restarting  
incoming text, 87

serial number on each  
disk, 114

set up for printer, see  
*printer setup*

SHIFT, use of, 16

Sider, 47

Silver Reed EXP-500  
printer, 69

SIN, 52

slot, 4, 7-8

Slot 1, 4

Slot 2, 4

Slot 3, 4, 7-8

Slot 6, 4

Slot-Buster Card, 66

Smith-Corona D-100, D-200,  
D-300 printers, 69

soft switches, 115-116  
locations of, 116

software theory of operation,  
113-115

Solid-Apple key, 9, 24

solid-circle label, 9

SP%, 64-65

space, also see *room*

effect on word wrap, 16

illus., 17

to get to open area on  
screen, 30

use of, 17

Spectrum JX-80, LX-80  
printers, 69

speech synthesizer, 88

speed of operation,  
importance of, 112

Spinwriter, 71

splitting a text, 48-50

Star Micronics SB-10, SD-10,  
SD-15, SG-10, SG-15,  
SR-10, SR-15, 69

start bit, 83

starting something of  
your own, 13-14

static electricity, 7

stop bit, 83-85

stopping, see

*emergency stop*

strange case, a, 50

subtraction, 52

Super Serial Card,  
3, 79, 83

black block, 79

switches, 79-80

switches, 66-67, 79-80

SwyftCard

conversion of SwyftCard

disks for use without

SwyftCard, 12

equipment, 3

installation, 7-8

restarting, 47-48,  
91-92

SwyftDisk, 5

SwyftWare

how to learn it, 2

installation of, 7-8

and other software, 47-48

problems with, 12

what it is, 1

and word processing, 1

software

how to run other, 47-48

ProDOS converter,  
102-110

SYNTAX ERROR, 89

## T

**TAB**  
command number, 123  
inserts spaces, 20  
key, 9  
and LEAP AGAIN, 27  
stops, 20

**TAN**, 52

tap a key, definition of, 16

target, 25

target character, 25

telephone, 82, 87

**Text**  
capacity, 39  
definition of, 15  
loading new, 41, 43-44  
saving first time, 40  
splitting, 48-50  
switching, 41  
when full, 48, 86-87

**text**  
definition of, 15  
structure of, 113

text file, ProDOS, 102-110

theory of operation  
hardware, 115-116  
software, 113-115  
user interface, 111-113

tokenized, 115

top margin, 61

top of keys, 9

Toshiba 1830, 72

track number, 97

transcendental functions, 52

transmission, 90

transplanting text,  
see *copy up*

Transtar 120, 73

tri-state driver, 116

turning off computer to get  
fresh start, 10

tutorial, 10-11  
beeps while doing, 11  
contents, 11  
fresh starts for, 10  
how to begin disk  
version, 10  
how to use disk  
version, 10

Type'n'Talk speech  
synthesizer, 88

typing  
capitals, 16  
in the middle, 17  
highlighting what you  
just typed, 32-33  
what it is, 15  
when you can, 15  
where it will appear, 15

## U

UE\$, 76-77

underline character,  
18-19

underlining  
at beginning of line,  
18-19  
continuous, 18-19  
how to, 18-19  
illus., 19  
last word in sentence, 18  
message, 76-77  
not shown on screen,  
18-19  
settings, 76-77  
single character, 18  
to make forms, 18-19

undo  
a deletion, 37  
a leap, 26-27  
typing, 35-36

uniformity of text, 113

unwrap, 113

uppercase letters,  
typing of, 16

US\$, 76-77

**USE FRONT**  
commands activated  
by, 20  
key, 9  
use of, 20

user interface theory of  
operation, 111-113

## V

verse on disk don'ts, 39

video monitor, 3, 5

voltage comparator, 115

voltages, 7

Votrax Inc., 88

Votrax speech  
synthesizer, 88

## W

warm start, 115

### WARNING

assigning command  
to G, 124

BASIC causing too  
many page breaks, 56

overriding DISK  
command, 92

turning off computer to  
start over, 12

using DISK command  
after CALL 3600, 12

What You See Is What  
You Get, 112-113

whistles, see *bells*  
and *whistles*

wide cursor, see *cursor*

width of a line of print, 60

WI%, 60

relation to MA%, 61

windblown desert

writing in sand on, 39-40

word processing  
and SwyftWare, 1

word wrap

definition of, 16

implementation, 113

spaces in, 16

write-protection, 41-42

write-protect tab, 41-42

WYSIWYG (pronounced  
"whiz-ee-wig"), see *What You  
See Is What You Get*

## X

Xerox Advantage D25, 73

XON-XOFF, 89-90

## Y

Yeti, see *Himalayas*

## Z

zucchini, 99

## Numerals and Symbols

?, see *question mark*

80-column card, 3

6551 chip, 85

27128, see *PROM*

10-4, that's a big